

**“Año de la Diversificación Productiva y del Fortalecimiento de la Educación”**

## **UNIVERSIDAD NACIONAL DE PIURA**



### **FACULTAD DE CIENCIAS**

#### **“DISEÑO DE UNA APLICACIÓN DE VOTO ELECTRÓNICO PRESENCIAL EN PLATAFORMA JAVA PARA SU IMPLEMENTACIÓN EN ANDROID”**

**Tesis para optar el título de Ingeniero Electrónico y Telecomunicaciones, que  
presenta el Bachiller:**

**SAMARA HASSAN DELGADO KHALIL**

**Asesor: ING. FRANKLIN BARRA ZAPATA**

**PIURA, JULIO DE 2015**

## **DEDICATORIA**

A mis padres Wasila y Guillermo, y a mi novia Olga, que están siempre a mi lado apoyándome e impulsándome a ser un mejor profesional y una persona de éxito, con su amor y ánimos.

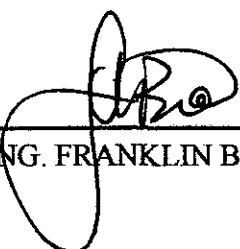
A los profesionales, especialmente de Ingeniería Electrónica y Telecomunicaciones, que son la esperanza de un mundo mejor, que con su empuje y dedicación son los representantes de un Perú en crecimiento.

## **ÍTULO DE LA TESIS**


**DISEÑO DE UNA APLICACIÓN DE VOTO ELECTRÓNICO PRESENCIAL EN  
PLATAFORMA JAVA PARA SU IMPLEMENTACIÓN EN ANDROID**

## **RESPONSABLES DEL DESARROLLO DE LA TESIS**

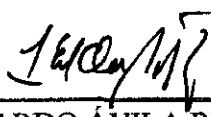
  
BACH. SAMARA HASSAN DELGADO KHALIL

  
ASESOR: ING. FRANKLIN BARRA ZAPATA

## **URADO EVALUADOR**

  
ING. JUAN VICENTE SANDOVAL  
PRESIDENTE

  
ING. CARLOS ARELLANO RAMÍREZ  
SECRETARIO

  
ING. EDUARDO ÁVILA REGALADO  
VOCAL

## **AGRADECIMIENTOS**

A Dios, por permitirme llegar a estas instancias y haberme iluminado durante el largo camino de mi carrera profesional para poder ser parte del futuro peruano y del mundo.

A mi asesor, Ing. Franklin Barra Zapata, por su apoyo incondicional y el aporte de sus conocimientos para que esta investigación se haya podido desarrollar de la mejor manera posible.

A mis padres Wasila y Guillermo, y a mi novia Olga, que fueron los principales participantes en la carrera para conseguir mis objetivos, con su dedicación, consejos y su amor incondicional.

## **RESUMEN**

Actualmente en nuestro país el proceso electoral se realiza de manera presencial llenando una cédula, la cual es colocada en una urna y luego, a través de delegados, se hace el conteo de votos manualmente. Generalmente, en los procesos electorales se presentan problemas de confiabilidad en los resultados dada la posibilidad de manipulación de éstos, además de los problemas que se relacionan con la obtención de los datos en las zonas rurales de nuestro país, por las distancias y la accesibilidad.

Aprovechando las tecnologías actuales como las tabletas y la facilidad que proponen los sistemas operativos como Android para ser programados y para crear aplicaciones, en este proyecto se presenta el desarrollo de una aplicación en dicho sistema operativo con lenguaje de programación Java, la cual será ejecutada desde una tableta para el proceso de votación y almacenará los resultados finales en una base de datos, para que, después del proceso electoral, se transmitan esos resultados vía bluetooth a un hardware diseñado con un microcontrolador PIC que estará conectado con un datalogger para la obtención y almacenamiento de los datos en una memoria USB.

Durante la explicación se analizará las ventajas que ofrece la aplicación y el diseño del hardware así como los requerimientos de los dispositivos a utilizarse. Y finalmente, se presentarán resultados del prototipo real, unas breves pero significativas conclusiones y se darán algunas recomendaciones y vistas a mejoramientos futuros.

## **ABSTRACT**

Currently in our country the electoral process is conducted in person by filling out a card, which is placed in an urn and then the vote count is done manually by delegate's hands. Generally, in the electoral process reliability issues are presented in the results due to possible manipulation of these, in addition, there are also problems related to data collection in rural areas of our country caused by the distance and accessibility to it.

Taking advantage of the existing technologies like tablets and the facility proposed by operating systems like Android to allow the programming and the creation of applications, it is presented in this project the development of an application in that operating system with Java programming language, which will be executed within a tablet for the voting process and the final results will be stored in a database, so that after the elections, these results are transmitted via Bluetooth to a hardware designed with a microcontroller PIC so as to be connected to a datalogger to obtain and storage the final data on a USB stick.

During the explanation of this project, the advantages of the application and the hardware design and the requirements of the to-be-used devices will be discussed. Finally, the results of a real prototype, brief but meaningful conclusions and some recommendations and views of future improvements will be detailed.

## ÍNDICE

<b>I. Introducción .....</b>	<b>9</b>
<b>II. Planteamiento Metodológico .....</b>	<b>11</b>
2.1 Realidad problemática.....	11
2.2 Definición y delimitaciones del problema.....	12
2.3 Formulación del problema .....	14
2.4 Objetivos del proyecto .....	14
2.4.1 Objetivo principal.....	14
2.4.2 Objetivos específicos.....	14
2.5 Justificación e importancia.....	15
2.6 Viabilidad y limitaciones de la investigación.....	15
2.6.1 Limitaciones .....	16
2.7 Hipótesis.....	16
2.8 Metodología .....	17
2.9 Procesamiento, análisis e interpretación de datos .....	17
2.10 Organización del proyecto .....	18
<b>III. Marco Teórico .....</b>	<b>19</b>
3.1 Elecciones.....	19
3.2 Voto electrónico .....	19
3.3 Modalidades del voto electrónico .....	20
3.4 Breve reseña histórica .....	21
3.4.1 Principios.....	21
3.4.2 Europa y Asia .....	22
3.4.3 Estados Unidos.....	22
3.4.4 Latinoamérica.....	23
3.5 Equipos de votación electrónica implementados .....	24
3.5.1 Urna electrónica en Brasil .....	24
3.5.2 SmartMatic SAES-3370 .....	25
3.5.3 SmartMatic en Venezuela .....	26
3.5.4 Perú.....	28
<b>IV. Diseño de la Aplicación - Software .....</b>	<b>29</b>
4.1 Software a utilizar .....	29
4.1.1 Android.....	29
4.1.2 Tableta.....	31

4.1.3 Java.....	32
4.1.4 Eclipse.....	32
4.2 Análisis del proyecto.....	33
4.3 Aplicación de votación.....	34
4.3.1 Tableta como cédula.....	35
4.3.2 Algoritmo de la aplicación.....	36
4.3.3 Diagrama de flujo de la aplicación.....	36
4.3.4 Diseño de la aplicación.....	38
4.4 Aplicación de envío de datos.....	51
4.4.1 Diagrama de flujo.....	51
4.4.2 Actividad de la aplicación.....	52
4.5 Resumen.....	55
<b>V. Diseño de la Aplicación - Hardware.....</b>	<b>56</b>
5.1 Componentes para el diseño.....	56
5.1.1 Microcontrolador PIC.....	56
5.1.2 Bluetooth.....	57
5.1.3 Datalogger.....	58
5.1.4 Otros componentes electrónicos.....	58
5.2 Análisis del proyecto.....	58
5.3 Algoritmo.....	59
5.4 Diagrama de flujo.....	60
5.5 Diseño.....	60
5.5.1 Módulo Bluetooth.....	62
5.5.2 Módulo Datalogger.....	63
5.5.3 Código del microcontrolador.....	64
5.6 Resumen.....	69
<b>VI. Prototipo y su Funcionamiento.....</b>	<b>70</b>
6.1 Votación.....	70
6.2 Envío de los datos.....	73
6.3 Costos.....	76
<b>VII. Conclusiones.....</b>	<b>77</b>
<b>VIII. Recomendaciones.....</b>	<b>79</b>
<b>Referencias.....</b>	<b>80</b>
<b>Anexos.....</b>	<b>83</b>



# CAPÍTULO I

## INTRODUCCIÓN

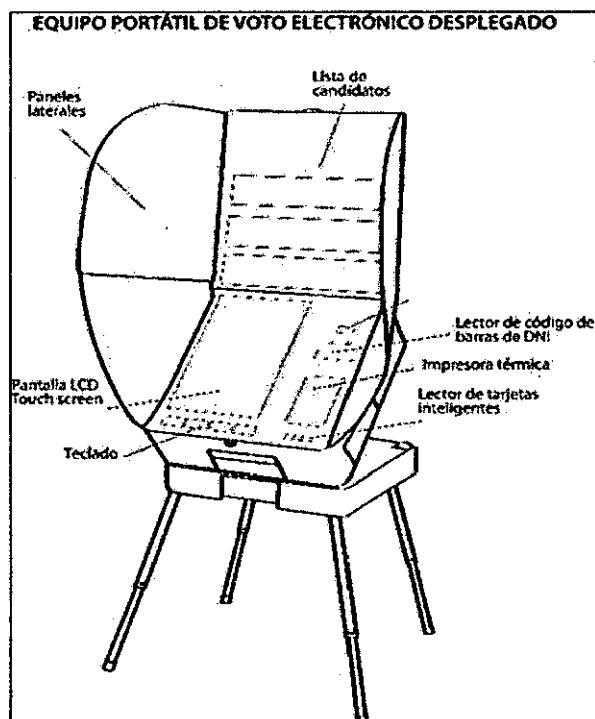
En los países democráticos, como el Perú, los gobernantes son seleccionados a través de los procesos electorales.

Estos procesos en el Perú, se realizan en un día establecido cada 5 años, donde los ciudadanos acuden a centros educativos que están organizados por mesas de votación. El elector recibirá una cédula y marcará el representante de su preferencia, depositará en un ánfora el sobre con su voto. Después del proceso de votación un jurado elegido por las autoridades hará el conteo de los votos y serán enviados a la ONPE para su procesamiento.

Un sistema de voto electrónico es importante porque permite reducir o minimizar los impactos Políticos o sociales, temporales y ecológicos. Pero los actuales sistemas, no permiten reducir el impacto económico generado tras la construcción de las máquinas electrónicas, debido al uso de procesadores y computadores que conducen a que dichas máquinas sean de dimensiones considerables. Esto es, los gastos para la manufacturación sobrepasan los millones. Un claro ejemplo es EE.UU. [1], donde hay mucho en juego: miles de millones de dólares en ventas de máquinas y el restablecimiento de la confianza pública en el proceso electoral. Estados Unidos espera que el 75% de los electores voten por medios electrónicos para el año 2015.

En algunos países latinoamericanos como Brasil y Venezuela [2], se ha hecho uso de las tecnologías actuales para elaborar y construir máquinas especiales basadas en microcontroladores y computadores para hacer del proceso electoral una *Votación Electrónica Presencial*.

En Perú, ya se ha implementado un prototipo de sistema de voto electrónico, que consta de una máquina robusta que tiene un lector de código de barras para acceder con el DNI y hacer la votación [2]. Actualmente la Oficina Nacional de Procesos Electorales (ONPE) posee una solución tecnológica para voto electrónico presencial, la cual se encuentra en la etapa de prototipo.



**Figura 1** Prototipo de máquina de voto electrónico ONPE 2012 [3]

En la Figura 1 se puede apreciar el prototipo creado por la ONPE para el diseño y construcción de máquinas de votación electrónicas. Cabe resaltar que estas máquinas poseen la desventaja de sus dimensiones, lo que las hace costosas y difíciles de transportar. Si pensamos en aquellas zonas rurales y lejanas de nuestro país donde el acceso es complicado y las condiciones ambientales impiden el transporte de grandes equipos, nos hace preguntar ¿Cómo vamos a distribuir la opción de voto electrónico en todas las zonas del Perú?

Es así pues que en este proyecto se plantea el diseño de una aplicación para uso de las tabletas como medio y/o máquina de votación. Gracias a sus dimensiones reducidas y su liviandad puede ser transportado fácilmente a cualquier sitio. Además se presentará el diseño de un hardware con un *microcontrolador PIC* con un módulo bluetooth que se comunicaría con la tableta para el traspaso de los datos a una base de datos que sería almacenada en una memoria USB a través de un módulo datalogger conectado también a tal microcontrolador.

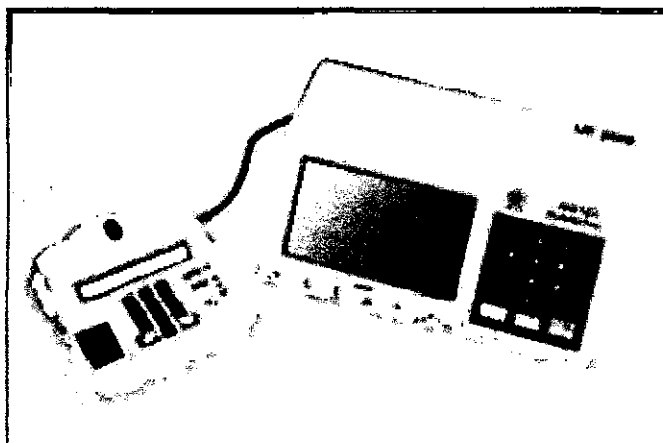
## CAPÍTULO II

### PLANTEAMIENTO METODOLÓGICO

#### 2.1. REALIDAD PROBLEMÁTICA

Actualmente en el Perú, los procesos electorales para elegir las autoridades representantes son llevadas a cabo cada 3 a 5 años. En esta modalidad, el voto es presencial, el votante se dirige hacia las urnas establecidas en centros como colegios e institutos, recibe una ficha o cédula con las imágenes representativas de los postulantes y marca su elección.

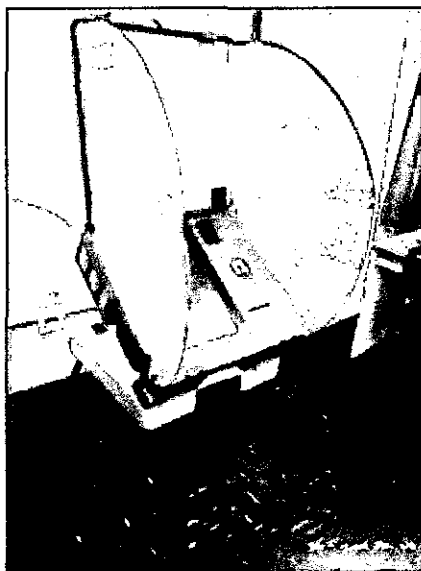
En otros países con mayor interés en la facilidad y reducción de los costos de los procesos electorales como EE.UU., Bélgica y España, ya se han implementado diversas tecnologías en base a computadoras [1] para realizar votaciones electrónicas. Por otro lado, en Latinoamérica, en Brasil (Ver Figura 2) y Venezuela, ya se ha implementado un sistema de voto electrónico, el cual permite al votante el uso de las herramientas tecnológicas de la información y herramientas modernizadas para realizar el proceso.



**Figura 2** Equipo de voto electrónico en Brasil [2]

En el Perú se ha elaborado un prototipo de un sistema de votación electrónica presencial (Ver Figura 3), la cual fue diseñada por personal de la Pontificia Universidad Católica del Perú y fue implementada por personal de la Universidad Nacional Mayor de San Marcos, basados en el prototipo ONPE 2012 (Ver Figura 1).

Este sistema consiste en una máquina especial y de grandes dimensiones que permite a través de un lector de código de barras para DNI, acceder a una pantalla donde se presenta la cédula de votación. Al final el votante recibe un ticket donde consta su votación.



**Figura 3** Prototipo Sistema de Voto Electrónico en Perú [4]

Observando estos avances para los sistemas de votación electrónica, nace este proyecto, que tiene como desarrollar un sistema de voto electrónico sobre un sistema operativo Android, de forma que se reemplace la fabricación de máquinas basadas en computadoras que son de dimensiones considerables y costosas, por tabletas, las cuales proporcionarán los medios para la realización del proceso electoral conjuntamente con una pequeño hardware basado en un microcontrolador PIC para la obtención de los datos y su almacenamiento.

## **2.2. DEFINICIÓN Y DELIMITACIONES DEL PROBLEMA**

Con el sistema actual de votación presencial que tenemos en Perú, surgen 4 problemas: político, económico, cronológico y de acceso. Los cuales, según [5], se pueden comprender como:

- ✓ La posibilidad de conocer con mayor prontitud los resultados electorales.
- ✓ La eliminación del fraude electoral.

- ✓ La reducción del tiempo de escrutinio y cómputo.
- ✓ La facilitación del ejercicio de voto por parte de los analfabetos.
- ✓ La posibilidad de realizar votaciones con gran frecuencia y a bajo costo.
- ✓ La comodidad para el votante.

*Político.* Debido a la cantidad de acusaciones de fraude y/o manipulación de los datos en las cédulas electorales, los cuales disminuyen la confianza y ponen en duda la legitimidad del proceso. Un claro ejemplo se ve en la controversia generada en las elecciones presidenciales del año 2000 [6] donde se acusó de manipulación de votos para favorecer a cierto candidato político.

*Económico.* Por el gasto generado para la adquisición de los materiales, como el papel para las cédulas de sufragio, los lapiceros, tinteros para marca de dedos, gastos de tinta de impresora y las mismas impresoras para la elaboración de las cédulas. Por ejemplo, el Jurado Nacional de Elecciones (JNE) sustentó su presupuesto de funcionamiento para el año 2015 de S/48'379,076 y presentó un requerimiento adicional de S/43'400.000 para atender las elecciones municipales complementarias de julio próximo y las actividades preliminares de las elecciones generales del año 2016 [7].

*Cronológico.* Ya que el tiempo para que el sistema del ONPE de resultados finales, 100% conteo de votos, es a veces hasta de una semana.

*De acceso.* En ciertas zonas rurales como el campo y la Amazonía del Perú no se puede acceder con facilidad para dejar los materiales y cédulas debido a las condiciones de las tierras, las condiciones ambientales, etc. En el Perú la actividad electoral ha sido bastante intensa en los últimos años; sin embargo, se ha escrito poco al respecto y menos aún, teniendo en cuenta al ciudadano del campo, y precisamente porque suelen venir de localidades más o menos distantes, los ciudadanos se preparan para esa jornada desde semanas anteriores y se desplazan en grupos visibles a lo largo de los caminos de la sierra o de los ríos del llano selvático [5]. Más aún cuando se han diseñado ya sistemas de votación electrónica como la mostrada en la Figura 1, donde el tamaño juega un papel importante en el transporte y considerando que actualmente se lucha por la inclusión e igualdad de toda la sociedad.

## **2.3. FORMULACIÓN DEL PROBLEMA**

Habiendo investigado y visto las limitaciones de los sistemas actuales de votación incluidos los sistemas de voto electrónico en Perú, surge la pregunta ¿Es posible reducir o eliminar los problemas actuales en los procesos electorales, utilizando las tecnologías existentes, para implementar un sistema de voto electrónico presencial, que sea confiable, económico, que permita el rápido escrutinio y entrega de resultados, y que pueda incluirse en los procesos electorales en todo el Perú?

## **2.4. OBJETIVOS DEL PROYECTO**

### **2.4.1. OBJETIVO PRINCIPAL**

Diseñar una aplicación que permita la votación electrónica, mostrando la cédula de votación en la pantalla de una Tableta con sistema operativo Android, donde se realice el voto y los datos se almacenen en la memoria, y se haga el procesamiento de los resultados (conteo). Diseñar un hardware para que al finalizar el proceso electoral se envíen los datos por comunicación Bluetooth a un microcontrolador PIC que a su vez los guarde en una memoria USB a través de un datalogger, y así transportar la información a los centros principales del organismo encargado de dicho proceso electoral.

### **2.4.2. OBJETIVOS ESPECÍFICOS**

- ✓ Crear una aplicación en Java para Android, usando eclipse. Dicha aplicación se encargará de identificar al votante, guardar su selección, y hacer el conteo de votos por candidato.
- ✓ Hacer el diseño de un hardware que a través de un microcontrolador de bajo costo como un PIC sea capaz de recibir los datos procesados y almacenarlos en memoria.
- ✓ La aplicación se diseña para la tecnología de una tableta como formato de votación y almacenamiento de datos, reemplazando las cédulas físicas y reemplazando el uso de computadores.

- ✓ Que los procesos electorales proporcionen al votante o ciudadano mayor confianza y credibilidad en los resultados finales, disminuyendo la posibilidad de fraude electoral y acelerar el proceso de conteo de votos.

## **2.5. JUSTIFICACIÓN E IMPORTANCIA**

La importancia de este proyecto radica en la utilización de las tecnologías actuales para mejorar el desarrollo de los procesos electorales en nuestro país.

Con estas tecnologías podremos mejorar los recursos y solucionar los problemas generados por la incertidumbre del votante ¿Se elegirá con justicia a nuestro Presidente y autoridades regionales? Además con los diseños propuestos por este proyecto se puede disminuir los costos y gastos de los procesos. Se pretende pues que en el futuro se utilicen tabletas y hardware desarrollado con microcontroladores para cumplir con los mejoramientos requeridos.

Otro punto importante y a destacar es el impacto ambiental que generaría el uso de aplicaciones en dispositivos electrónicos, ya que se evitaría el uso de papel y tinta. La mejora del proceso conlleva a la reducción del tiempo de presentación de los resultados, algo que garantiza la aceptación y confianza del pueblo. Y finalmente, el desarrollo de una aplicación en una tableta facilita el transporte de la misma a las zonas menos accesibles de nuestro país.

## **2.6. VIABILIDAD Y LIMITACIONES DE LA INVESTIGACIÓN**

Se ha evaluado este proyecto según los problemas descritos anteriormente y se concluye que es un proyecto viable, de mejoras económicas, sociales y en accesibilidad. Los requerimientos para este proyecto incluyen software accesible en el mercado y en internet, tales como libros de programación, blogs, libros virtuales; y conocimientos de programación y simulación de hardware.

### 2.6.1. LIMITACIONES

- ✓ Acceso a la información sobre programación y codificación para Android y para microcontroladores PIC, lo que requiere que el desarrollador del proyecto emplee lógica y tenga conocimientos previos de programación en Java.
- ✓ El proyecto llegará hasta el diseño de la aplicación para votación electrónica y para la transferencia de los datos por bluetooth, así como el diseño de un hardware con un microcontrolador PIC que a su vez los almacenará en una memoria USB usando un datalogger. Esto requiere conocimiento de programación y simulación electrónica, a su vez la comprensión de comunicación RS232 o serial para la configuración del módulo Bluetooth y Datalogger.

### 2.7. HIPÓTESIS

Los sistemas y equipos actuales de votación en Latinoamérica y en especial en Perú, generan incertidumbre y duda en los ciudadanos con respecto a la credibilidad y veracidad del proceso. Tales sistemas poseen características no favorables en los ámbitos económico y político.

En Perú ya se ha creado un prototipo de máquina para incentivar el desarrollo del voto electrónico en nuestro país. Dicho prototipo consiste en el uso del computador como medio de votación, almacenamiento, procesado y envío de la información, lo que hace que la máquina sea robusta, pero de dimensiones grandes y difícil para su transporte a todas las poblaciones de nuestro país.

Al analizar esta realidad problemática y los objetivos planteados en este proyecto, se genera la hipótesis:

***La utilización de software y hardware como Android y tabletas, así como la electrónica digital, podrán ayudar al desarrollo de un nuevo sistema de voto electrónico que minimice o elimine las desventajas que poseen los sistemas actuales, incluidos aquellos sistemas de voto electrónico basados en computadoras.***



## **2.8. METODOLOGÍA**

De acuerdo con el propósito de este proyecto, la naturaleza de la problemática y los objetivos formulados, la investigación será de carácter técnico y de diseño. Este tipo de investigación tiene como finalidad la solución de problemas prácticos, lo cual implica intervención o modificación de la propia realidad; se pretende demostrar la utilidad de la intervención que se realizará y aplicarla en pequeña escala.

El desarrollo de la presente investigación se basará en el método de la ciencia o, simplemente, método científico; el mismo que está conformado por una serie de actividades correctamente articuladas que conducen a un resultado coherente con la propuesta de solución.

De manera general, tales actividades son: problema nuevo para la ciencia, objetivos de la investigación, hipótesis de la investigación, variables, universo y muestra, técnicas e instrumentos, aplicación de los instrumentos, análisis de la información, contrastación de la hipótesis, formulación de conclusiones y recomendaciones.

## **2.9. PROCESAMIENTO, ANÁLISIS E INTERPRETACIÓN DE DATOS**

- ✓ Estudio de los antecedentes.
- ✓ Estudio y definición de conceptos.
- ✓ Estudio de los sistemas de votación presenciales y electrónicos en Latinoamérica y nuestro país.
- ✓ Selección del lenguaje de programación Java para desarrollo de la aplicación en Android.
- ✓ Selección de los componentes electrónicos para el diseño del hardware.
- ✓ Técnicas de programación y habilidades para diseño de la aplicación.
- ✓ Pruebas en prototipo real.

## **2.10. ORGANIZACIÓN DEL PROYECTO**

Esta tesis está compuesta de ocho capítulos. En el Capítulo I y II se presenta una breve introducción a la realidad problemática, los objetivos, y las razones para el desarrollo de este proyecto, así como el análisis y justificación del mismo. En el Capítulo III se presentará un compendio de todo el marco teórico que comprende el desarrollo de la tesis, es decir, antecedentes y definiciones de la votación electrónica en Latinoamérica y en el Perú; y presentando también los diferentes prototipos y diseños ya elaborados a lo largo de los últimos años, su composición, sus elementos electrónicos, etc. En el Capítulo IV, el diseño de la aplicación en Android será detallada, haciendo hincapié en los puntos claves del diseño y presentando los procedimientos y decisiones tomadas, así como se ilustrará las observaciones pertinentes de dicho diseño. El Capítulo V estará dedicado a explicar todo lo relacionado al diseño del hardware, donde se indicará los componentes seleccionados y los métodos usados para la programación, diseño y los protocolos de comunicación entre el microcontrolador y los diferentes módulos. En el Capítulo VI se presentará el prototipo real elaborado a partir de los diseños presentados en los capítulos precedentes, donde se mostrarán fotos e imágenes del funcionamiento final del proyecto. Finalmente en los Capítulos VII y VIII se detallan las conclusiones de este proyecto, resumiendo las especificaciones y el comportamiento del diseño, así como las recomendaciones finales y las recomendaciones para mejorar el diseño en el futuro.

## CAPÍTULO III

### MARCO TEÓRICO

Para poder entender bien los procedimientos relacionados a la votación electrónica y luego poder explicar el desarrollo de la aplicación y hardware de este proyecto, empezaré por definir algunos conceptos básicos sobre votación, de manera general, para después poder entrar en detalle en la parte electrónica y los sistemas actuales que ya se han venido desarrollando en ciertas partes de Latinoamérica y el Perú.

#### 3.1. ELECCIONES

En política [8], una elección es un proceso de toma de decisiones en donde los ciudadanos votan por sus candidatos o partidos políticos preferidos para que actúen como representantes en el gobierno. Es la esencia y parte principal de un sistema democrático.

*Elecciones Municipales.* Modo democrático para elegir a Alcaldes y Regidores de los Concejos Municipales Provinciales y Distritales en toda la República.

*Elecciones Regionales.* Modo democrático para elegir al Presidente, Vicepresidente y miembros del Consejo Regional.

*Escrutinio.* Etapa del proceso electoral que consiste en la verificación, clasificación y conteo de los votos para registrar los resultados de una elección.

#### 3.2. VOTO ELECTRÓNICO

En [9] se define el voto electrónico como una forma de votación basada en medios electrónicos que se diferencia del método tradicional por la utilización de componentes de hardware y software que permiten automatizar los procesos de comprobación de la identidad del elector, emisión del voto, conteo (escrutinio) de votos,

emisión de reportes de resultados; así como de una red de comunicaciones para la transmisión y presentación de resultados de un proceso electoral.

Las tecnologías para el voto electrónico pueden incluir tarjetas perforadas, sistemas de votación mediante escáneres ópticos y quioscos de votación especializados (incluso sistemas de votación autocontenidos, sistemas de votación de Registro o Grabación Electrónica Directa, DRE por sus siglas en inglés). También puede referirse a la transmisión de papeletas y votos por vía telefónica, redes privadas de computación o por la Internet.

Las tecnologías del voto electrónico pueden acelerar el conteo de los votos y proveer una mejor accesibilidad para los votantes con algún tipo de discapacidad.

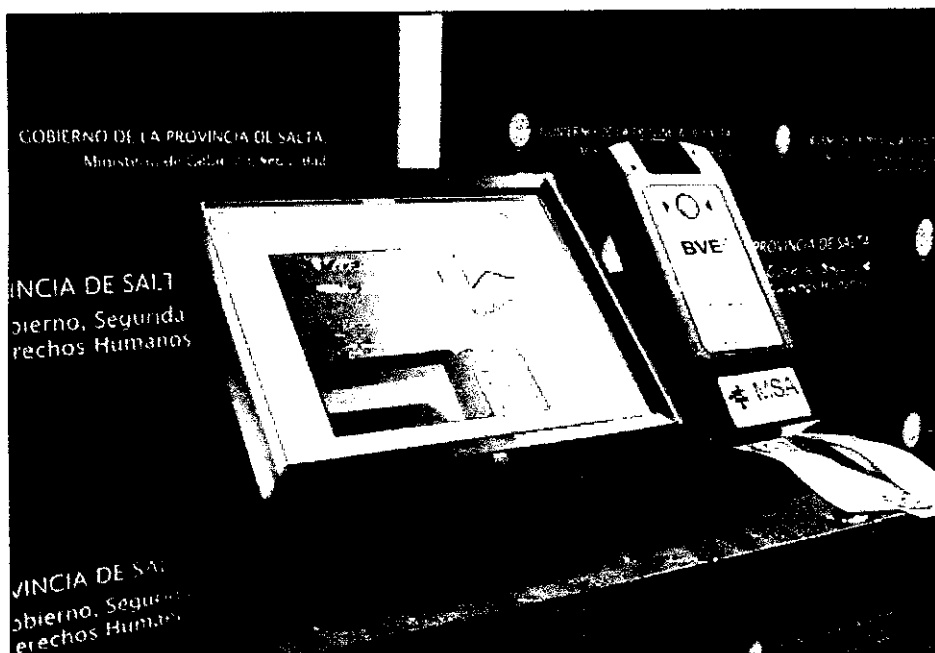
### **3.3. MODALIDADES DEL VOTO ELECTRÓNICO**

*Registro electrónico directo*, cuando se identifica manualmente al elector, autorizándolo a utilizar una máquina, que en este caso genéricamente se denomina DRE (direct recording electronic) o sistema de registro electrónico directo, dispuesta en un lugar específico (colegio electoral). En dicho caso, el proceso de identificación es independiente y existe la posibilidad de relacionarlo con el voto depositado [10].

*Urna Electrónica*, con un escáner se contabiliza el voto en el momento en que la papeleta entra en la urna. Al cierre de la mesa, cada urna contabiliza los votos e imprime el resultado, sin intervención humana. Los datos son enviados.

*Pantalla Táctil*, es una máquina de votación (Ver Figura 4) que presenta menús de selección en pantalla en los que el elector elige, valida el voto mediante una tarjeta de identificación, y el voto quedará almacenado en memoria para transmitirlo al cierre de la jornada electoral.

*Internet*, más complicada que las anteriores. Es necesario que cada votante disponga de una clave especial de identificación, que junto con su número de identidad, le den acceso al sistema, y ahí es donde se produce el principal problema [11].



**Figura 4** Ejemplo de modalidad de Pantalla Táctil [12]

### **3.4. BREVE RESEÑA HISTÓRICA**

#### **3.4.1. Principios**

Los sistemas de voto electrónico han sido utilizados desde épocas tan lejanas como los años 60.

En dichos años, los sistemas de tarjeta perforada fueron puestos en práctica, por primera vez, en el estado de Oregon, Estados Unidos. A pesar de que generaron problemas desde los años 60, no fue sino hasta las elecciones de noviembre del 2000 cuando la discusión sobre la exactitud de las "punch cards" tomó dimensión pública.

Durante los años 80 [13], diversos informes reseñaron la falta de exactitud en los recuentos en elecciones realizadas con los sistemas de tarjetas perforadas, y existieron importantes disputas (que llegaron a la Corte Suprema norteamericana) sobre las fallas en el software de los sistemas de lectura ópticos.

Ya en los 90, los sistemas de lectura óptica se habían convertido en una plataforma de recuento bastante sólida, pero fueron perdiendo "mercado" frente a los

nuevos sistemas de grabación directa electrónica, con pantallas táctiles, teclados, e interfaces informáticas.

### **3.4.2. Europa y Asia**

- ✓ Suiza, la mayor parte de los ciudadanos utilizan y confían en el voto electrónico, desde los referendos del 2003 y 2004.
- ✓ Holanda, el mayor ensayo se desarrolló en junio de 2004 para las elecciones al Parlamento Europeo.
- ✓ Inglaterra, en junio de 2004 se implementó una prueba de voto electrónico en Londres. El proceso seguido es minucioso y se desarrolla con tiempo, obteniendo de esta forma un avance seguro hacia un escenario de voto electrónico bien diseñado y correctamente planificado.
- ✓ Alemania, en septiembre de 2005, utilizó el voto electrónico presencial, para las elecciones parlamentarias de forma vinculante en algunos colegios, con éxito desigual.
- ✓ Francia, en los últimos comicios presidenciales de 2007 se emplearon urnas electrónicas por parte de 1,5 millones de ciudadanos de un total de 44,5 millones.
- ✓ España, en febrero de 2005 se implementó la primera prueba de voto electrónico por Internet, la misma que resultó ser un fracaso técnico y de participación.
- ✓ Italia, también utilizó el voto electrónico no vinculante y a pequeña escala en colegios electorales.
- ✓ También destacan Bélgica, Escocia, Irlanda, Austria, Eslovenia, Hungría, Estonia, Australia e India.

### **3.4.3. Estados Unidos**

En los comicios nacionales de los Estados Unidos celebrados en el año 2004, la mayor parte de los votantes se valió de sistemas automatizados; 13,7% de los ciudadanos sufragaron con tarjetas perforadas; 14% empleó sistemas similares a la manivela de hace más de 100 años; 34,9% votó en equipos de lectura óptica y 29,3% utilizó para sufragar equipos desarrollados bajo el concepto del Registro Electrónico Directo [14].

3.4.4. Latinoamérica

Venezuela

Se plantearon problemas muy interesantes debido a la sospecha de que se pudieran relacionar las listas de votantes al pasar en un determinado orden y el propio voto emitido en una DRE de la empresa Smartmatic, que se depositaba en orden secuencial. De forma que en las elecciones del año 2005 se retiraron cautelarmente, pero se volvieron a utilizar en diciembre de 2006. Se implementó el sistema electrónico al 100% a partir del año 2009, utilizando equipos de la empresa Smartmatic, destacando el modelo SAES 4000 [15].

Brasil

Brasil es el único país del continente americano que ha implementado satisfactoriamente un sistema de votación electrónica (Ver Figura 5) en una elección de carácter nacional. Para llegar a implementar el 100% de este sistema en su territorio nacional el Tribunal Superior Electoral (TSE) empezó gradualmente a automatizar los diferentes procedimientos que forman parte del sufragio y el escrutinio [16].

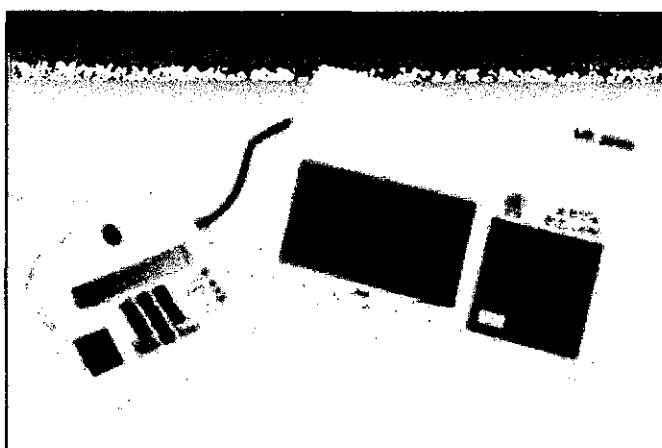
No automatizados	Baja automatización	Automatización intermedia	Alta automatización
Bolivia	Costa Rica	Chile	Argentina
Colombia	México	Guatemala	Brasil
Ecuador		Honduras	Paraguay
El Salvador		Perú	Venezuela
Nicaragua			
Panamá			
Puerto Rico			
Uruguay			

Figura 5 Grado de automatización del voto electrónico en América Latina [10]

### 3.5. EQUIPOS DE VOTACIÓN ELECTRÓNICA IMPLEMENTADOS

#### 3.5.1. URNA ELECTRÓNICA EN BRASIL

Como se mencionó antes, Brasil es el único país que ha implementado 100% un sistema de votación electrónica. En la Figura 6 se puede apreciar el equipo estándar utilizado en las elecciones de Brasil. Como se menciona en [2], la urna electrónica está compuesta por un terminal de mesa y un terminal de elector. En el terminal de mesa se identifica al elector a través de la huella dactilar o digitando los números de su documento de identidad, con lo cual se habilita el terminal del elector para realizar un voto, el elector digita los números de sus opciones políticas y verifica en la pantalla sus selecciones, de estar de acuerdo confirma su voto. Se imprime un comprobante del voto, el cual no es accesible al elector y que sirve para una auditoría posterior a la jornada electoral.



**Figura 6** Equipo de voto electrónico en Brasil [2]

El terminal del elector posee los siguientes componentes:

- ✓ Gabinete
- ✓ Placa madre
- ✓ Procesador compatible con X86 de 32 bits
- ✓ RAM socket DDR
- ✓ Reloj interno
- ✓ Generador de sonidos



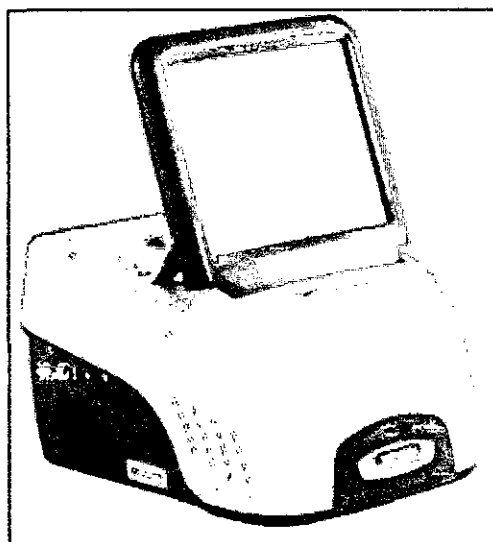
- ✓ Controlador de video VGA
- ✓ Controlador SATA, ATA 2
- ✓ Unidades de almacenamiento
- ✓ Teclado numérico
- ✓ Display LCD 10" 800x600
- ✓ Fuente de alimentación
- ✓ Alimentación externa con tensión de entrada variando entre 90 a 220 VAC
- ✓ Alimentación interna con batería de 12 voltios dc
- ✓ Batería interna
- ✓ Diodos emisores de luz
- ✓ Mecanismo de impresión

El terminal de mesa posee los siguientes componentes:

- ✓ Gabinete
- ✓ Display alfanumérico
- ✓ Display gráfico 2", 240x320 colores
- ✓ Teclado numérico
- ✓ Diodos emisores de luz
- ✓ Dispositivo de lectura de huellas dactilares
- ✓ Dispositivo de lectura y grabación de tarjetas inteligentes
- ✓ Generador de sonidos
- ✓ Puerto USB 2.0

### **3.5.2. SMARTMATIC SAES-3370**

Bélgica tiene una de las comisiones electorales más visionarias del mundo. Fue pionera del voto electrónico en los años 90, y desde entonces, siempre se ha mantenido a la vanguardia en la adopción de tecnología electoral. Las elecciones al Parlamento Europeo 2014 marcaron la primera vez en la historia en que los belgas utilizan un sistema de votación totalmente verificable para elegir a los miembros del Parlamento. Smartmatic facilitó la experiencia electoral de los municipios que contrataron nuestros servicios [17].



**Figura 7** Equipo SMARTMATIC SAES-3370 [18]

Este equipo tiene las siguientes características:

- ✓ Una máquina de votación táctil de última generación, con una pantalla de votación 17".
- ✓ 100% auditable. Los votos son registrados en la memoria electrónica redundante y en varias circunstancias diferentes.
- ✓ Votos cifrados usando algoritmos simétricos de 256 bits.
- ✓ Impresora integrada para los recibos de papel, informes y registros.
- ✓ Lector de tarjetas inteligentes para la activación de la sesión de votación.
- ✓ Capacidad de transmisión de datos segura a través de redes seguras unidireccionales (LAN, teléfono de línea fija, GSM, CDMA y satélite).
- ✓ Módulo de accesibilidad integrada para los votantes con discapacidad visual.

### **3.5.3. SMARTMATIC EN VENEZUELA**

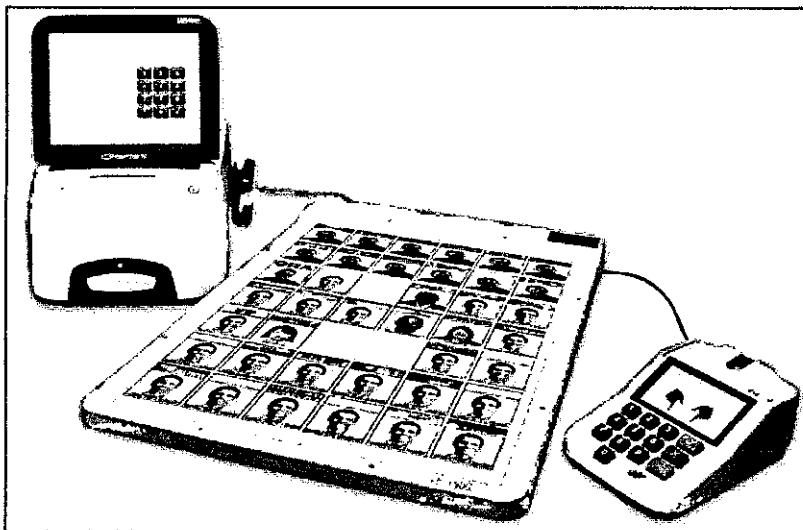
Los modelos utilizados desde el 2009 son SAES-3000, SAES-3300 y SAES-4000. El modelo SAES-4000 mostrado en la Figura 8 —el más usado— tiene las siguientes características:

- ✓ Monitor VGA de 10.4 pulgadas
- ✓ Pantalla táctil resistiva

- ✓ Microprocesador Celeron 600Mhz
- ✓ Memoria RAM máxima de 1 GB
- ✓ Dispositivo de almacenamiento (memoria interna de 256 MB, un slot para compact flash)
- ✓ Impresora térmica con ancho de papel de 8 cm.
- ✓ 6 puertos USB 2.0
- ✓ Un puerto PS2 para la boleta electrónica
- ✓ Un puerto de red RJ-45
- ✓ Peso del equipo igual a 5.9 Kg.
- ✓ Alimentación externa con tensión de entrada variando entre 100 a 240 VAC

**Accesorios:**

- ✓ Boletas electrónicas
- ✓ Batería externa
- ✓ Maleta protectora
- ✓ Botón de control remoto
- ✓ Activador de tarjeta Smartcard
- ✓ Dispositivo Capta-huella



**Figura 8 SMARTMATIC SAES-4000 [18]**

3.5.4. PERÚ

Se menciona en [3] que el sistema se compone por dos equipos que no tienen conexión física entre ellos. Uno permite la identificación del votante a través de la lectura del número de su DNI mediante un escáner, o digitando en la pantalla táctil los números del DNI y a la vez comprobando que el votante pertenece a la mesa de votación. Se activa una tarjeta que habilitará y permitirá realizar un solo voto en el otro equipo llamado cabina de votación. En la cabina de votación, el votante introduce la tarjeta en la ranura del lector de tarjetas, la aplicación de voto electrónico por la pantalla le presenta la cédula de votación para que pueda ejecutar un solo voto. Finalmente, el voto se imprime en papel térmico mostrando al votante su selección. En la Figura 3 presentada anteriormente se puede apreciar el sistema de la ONPE, y en la Figura 9 se aprecia el diagrama de bloques de su funcionamiento.

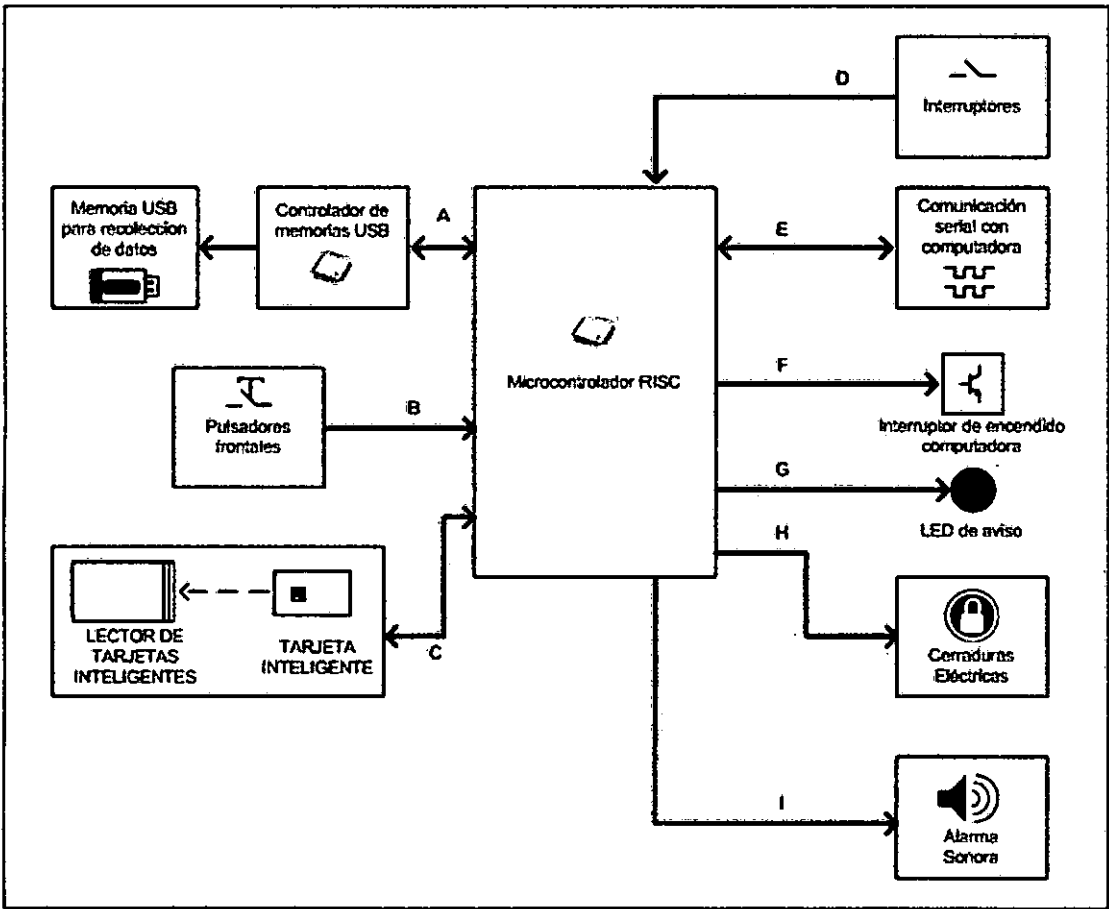


Figura 9 Diagrama de bloques del controlador de la máquina de votación Perú-ONPE

## CAPÍTULO IV

### DISEÑO DE LA APLICACIÓN – SOFTWARE

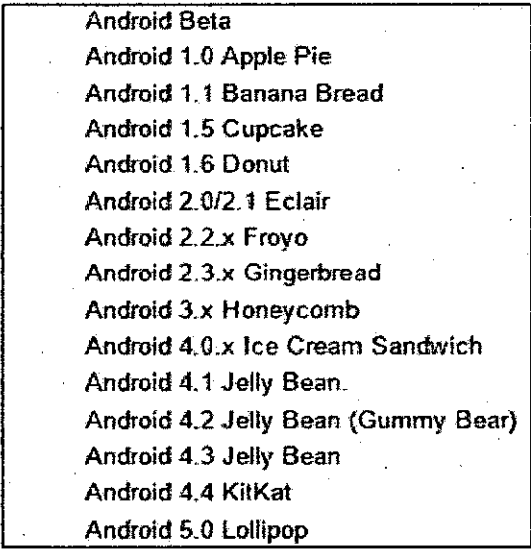
#### 4.1. SOFTWARE A UTILIZAR

##### 4.1.1. ANDROID

Android es un sistema operativo basado en Linux [19], diseñado principalmente para móviles con pantalla táctil como teléfonos inteligentes o tabletas. Android fue presentado en 2007 junto la fundación del Open Handset Alliance: un consorcio de compañías de hardware, software y telecomunicaciones para avanzar en los estándares abiertos de los dispositivos móviles. El primer móvil con el sistema operativo Android se vendió en octubre de 2008.

Tiene una gran comunidad de desarrolladores escribiendo aplicaciones para extender la funcionalidad de los dispositivos. A la fecha, se han sobrepasado las 700.000 aplicaciones disponibles para la tienda de aplicaciones oficial de Android: Google Play. Los programas están escritos en el lenguaje de programación Java.

En la Figura 10 se puede ver la lista de todas las versiones de Android.



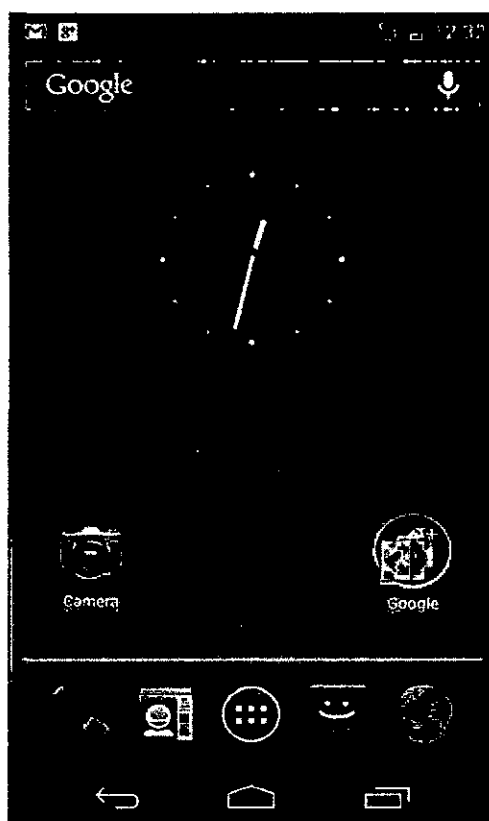
Android Beta
Android 1.0 Apple Pie
Android 1.1 Banana Bread
Android 1.5 Cupcake
Android 1.6 Donut
Android 2.0/2.1 Eclair
Android 2.2.x Froyo
Android 2.3.x Gingerbread
Android 3.x Honeycomb
Android 4.0.x Ice Cream Sandwich
Android 4.1 Jelly Bean
Android 4.2 Jelly Bean (Gummy Bear)
Android 4.3 Jelly Bean
Android 4.4 KitKat
Android 5.0 Lollipop

**Figura 10** Versiones de Android [19]

## Aplicaciones

Las aplicaciones se desarrollan habitualmente en el lenguaje Java con Android Software Development Kit (Android SDK), pero están disponibles otras herramientas de desarrollo, incluyendo un Kit de Desarrollo Nativo para aplicaciones o extensiones en C o C++, Google App Inventor, un entorno visual para programadores novatos y varios cruz aplicaciones de la plataforma web móvil marcos y también es posible usar las bibliotecas Qt gracias al proyecto Necesitas SDK.

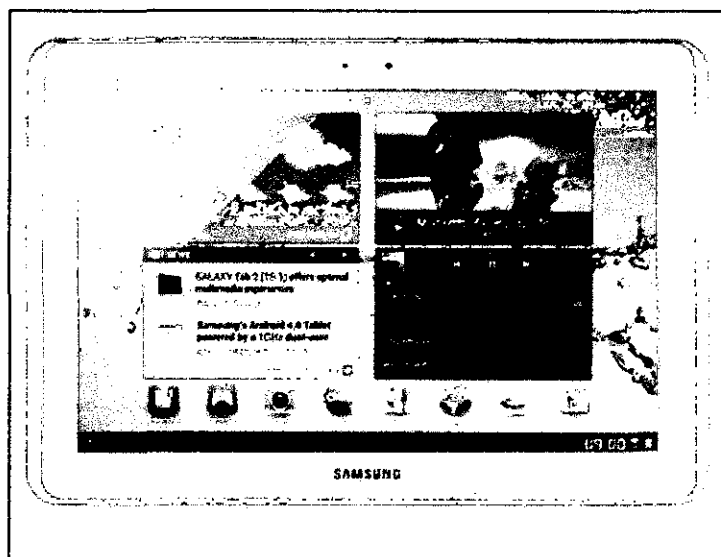
El desarrollo de aplicaciones para Android no requiere aprender lenguajes complejos de programación. Todo lo que se necesita es un conocimiento aceptable de Java y estar en posesión del kit de desarrollo de software o «SDK» provisto por Google el cual se puede descargar gratuitamente. Todas las aplicaciones están comprimidas en formato APK, que se pueden instalar sin dificultad desde cualquier explorador de archivos en la mayoría de dispositivos. En la Figura 11 se puede observar el entorno Android 4.1 instalado en un dispositivo tableta.



**Figura 11** Android 4.1 Jelly Bean [19]

#### 4.1.2. TABLETA

Una tableta [20] es una computadora portátil de mayor tamaño que un teléfono inteligente o una PDA, integrado en una pantalla táctil (sencilla o multitáctil) con la que se interactúa primariamente con los dedos o una pluma stylus (pasiva o activa), sin necesidad de teclado físico ni ratón. Estos últimos se ven reemplazados por un teclado virtual y, en determinados modelos, por una mini-trackball integrada en uno de los bordes de la pantalla. En la Figura 12 se puede apreciar el diseño de una tableta de la marca Samsung.



**Figura 12** Samsung Galaxy Tab 10.1 [20]

Aunque la tableta es un dispositivo, se considera en esta sección como parte del diseño software porque finalmente la aplicación diseñada puede ser instalada en una tableta para la aplicación de votación electrónica.

#### *Ventajas*

Las tabletas presentan ciertas ventajas sobre las computadoras, que ponen de manifiesto su utilidad para la elaboración y cumplimiento de objetivos de este proyecto. Éstas son:

- ✓ Su facilidad de uso en ambientes no favorables a un teclado y un ratón, como en la cama, de pie, o el manejo con una sola mano.
- ✓ Su peso ligero.
- ✓ El entorno táctil hace que en ciertos contextos el trabajo sea más fácil que con el uso de un teclado y un ratón.
- ✓ Facilita la realización de dibujos digitales y edición de imágenes pues resulta más preciso e intuitivo que pintar o dibujar con el ratón.
- ✓ Facilita y agiliza la posibilidad de agregar signos matemáticos, diagramas y símbolos.
- ✓ La duración de la batería es mucho mayor que la de una computadora portátil.

#### **4.1.3. JAVA**

Java es un lenguaje de programación y la primera plataforma informática creada por Sun Microsystems en 1995. Es la tecnología subyacente que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios. Java se ejecuta en más de 850 millones de ordenadores personales de todo el mundo y en miles de millones de dispositivos, como dispositivos móviles y aparatos de televisión.

El lenguaje [21], deriva mucho de su sintaxis de C y C++, pero tiene menos facilidades de bajo nivel que cualquiera de ellos. Las aplicaciones de Java son generalmente compiladas a bytecode (clase Java) que puede correr en cualquier máquina virtual Java (JVM) sin importar la arquitectura de la computadora. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo (conocido en inglés como WORA, o "write once, run anywhere").

#### **4.1.4. ECLIPSE**

Eclipse [22] es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. Esta plataforma, típicamente ha sido usada para desarrollar entornos de desarrollo integrados (del inglés IDE), como el IDE de Java



llamado Java Development Toolkit (JDT) y el compilador (ECJ) que se entrega como parte de Eclipse (y que son usados también para desarrollar el mismo Eclipse).

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios. En la Figura 13 se muestra la ventana principal de Eclipse con un programa en Java.

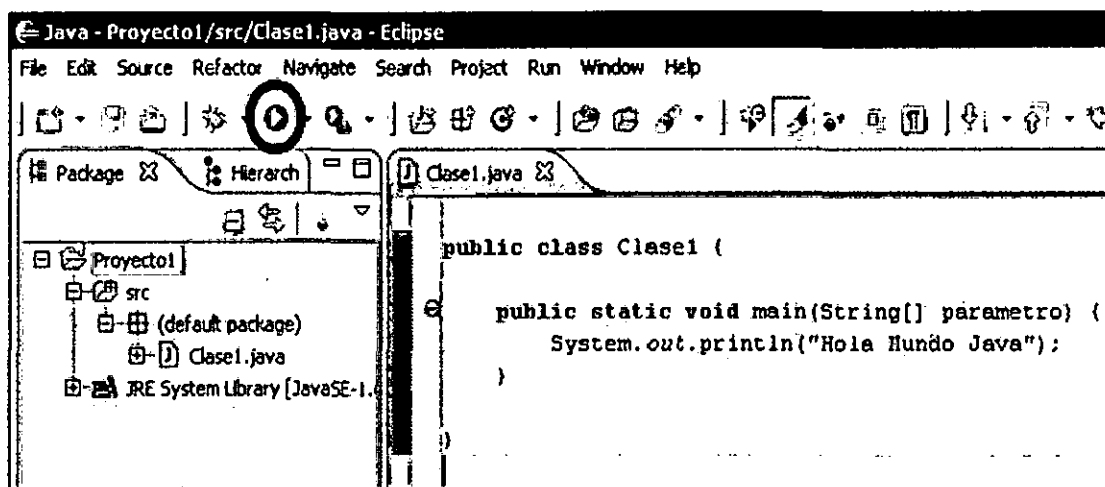


Figura 13 Ventana principal de Eclipse y programa en Java [23]

## 4.2. ANÁLISIS DEL PROYECTO

En este capítulo, donde se trata directamente con el diseño de la aplicación y software, se explicará las dos partes principales que tendrá el sistema de votación electrónica. Una de ellas es el diseño de la aplicación hablando de la parte de votación propiamente dicho, donde se diseñará el entorno que contendrá los íconos, botones, números; con lo que se podrá ingresar el DNI y hacer la selección del candidato que uno desee y, en una base de datos se almacenará los resultados. En la segunda parte, dentro de la misma aplicación se hará la comunicación bluetooth para el envío de los resultados a un hardware que los almacenará, con lo que luego podrían llevarse los resultados a las oficinas centrales.

### 4.3. APLICACIÓN DE VOTACIÓN

Actualmente, Android es el top en los sistemas que requieren controlar y manejar hardware como cámaras, celulares, tabletas, audio, video. La cantidad de herramientas y utilidades que encontramos en la web y libros permiten a los programadores acceso a las propiedades de este sistema operativo, su manipulación y programación en lenguajes ya conocidos como Java y C.

Esta parte del proyecto consiste en un sistema de voto electrónico diseñado para la plataforma Android. Se trata de crear una aplicación (software) en el programa Eclipse, que como se ha descrito anteriormente, permite la programación y compilación usando la IDE de JAVA. Dicha aplicación es un programa prototipo que permite realizar un proceso electoral como si lo hiciéramos en un ordenador, el usuario entrará a la aplicación realizará el voto y este voto será almacenado en la memoria de la tableta en una base de datos.

La aplicación tiene las siguientes características:

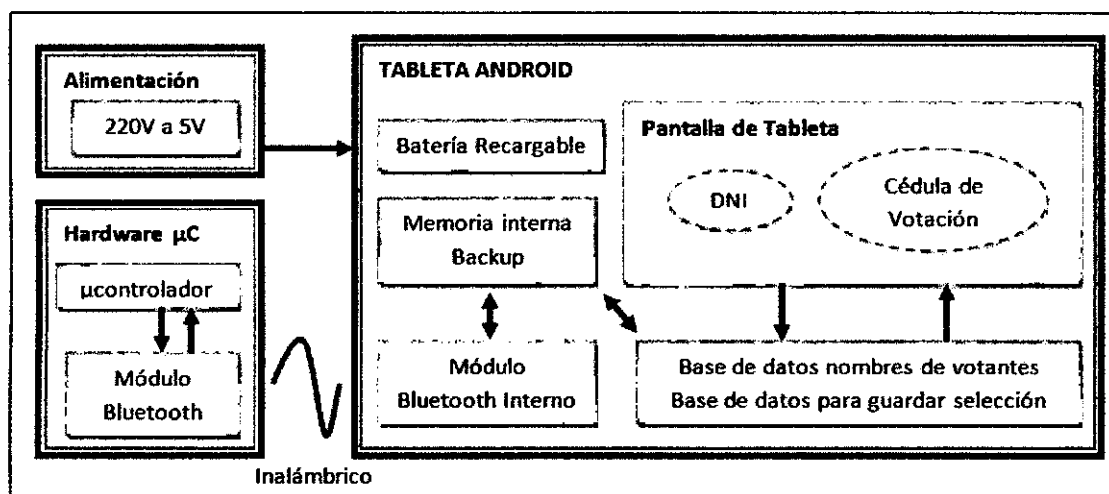
- ✓ Permite el ingreso de los dígitos del DNI. A través de un teclado numérico para mayor comodidad del votante.
- ✓ Reconoce si el usuario pertenece a esa mesa de votación y avisar si es que el usuario puede o no votar.
- ✓ Presenta en pantalla la cédula de votación con la imagen y nombres de los candidatos.
- ✓ Ventana de confirmación de voto. Asegura al votante que su elección ha sido registrada y que se ha almacenado como conteo en la base de datos.
- ✓ Voto único. No permite que un usuario ingrese luego de haber realizado su votación.
- ✓ Almacena en memoria interna o externa (micro SD) una base de datos donde llevará la cuenta de la cantidad de votos para cada candidato, y una base de datos donde registrará las personas que realizaron el voto y las que no.

### 4.3.1. TABLETA COMO CÉDULA

Como parte del diseño de la aplicación se considera este equipo principal para el sistema de votación electrónica. Una tableta que contará con sistema operativo Android y que a través de la aplicación será capaz de:

- ✓ Permitir la votación a través de su pantalla.
- ✓ Reconocer a través de base de datos si el usuario o votante pertenece o no a la mesa de votación.
- ✓ Almacenar la información de quién es el votante y la elección realizada de forma separada.
- ✓ Se alimentará a la red de 220V.

En la Figura 14 se aprecia el diagrama de bloques de lo que sería el funcionamiento de la tableta con la aplicación diseñada en Eclipse.



**Figura 14** Diagrama de bloques del diseño de la aplicación

La tableta puede ser de cualquier modelo o marca, pero tiene que cumplir ciertos requerimientos que, según lo investigado y dados los objetivos planteados en este proyecto, son los óptimos para lograr crear un sistema de votación electrónica confiable, económico y fácil de transportar. Así mismo debe contar con ciertas características básicas. Éstos son:

- ✓ Bajo Costo. Una tableta Android básica cuesta entre 199 a 399 nuevos soles.
- ✓ Cédula. La tecnología de la tableta permite diseñar la cédula de votación para ser presentada en la pantalla.
- ✓ Dispositivo ligero. De tamaño pequeño, poco peso y fácil de transportar.
- ✓ Registro del proceso. A través de la programación es posible almacenar a los votantes y sus votos en dos bases de datos diferentes en Android.
- ✓ Energía suficiente. Para el proceso electoral que es mayor de 10 horas, la alimentación es a 220V, adicionalmente en caso de emergencias la tableta cuenta con batería de litio que puede durar entre 10 a 14 horas.
- ✓ Memoria interna. Donde se pueda almacenar la base de datos.
- ✓ Base de datos ONPE. La tableta estará precargada con base de datos que permitirá saber si el votante pertenece o no a la mesa de votación.

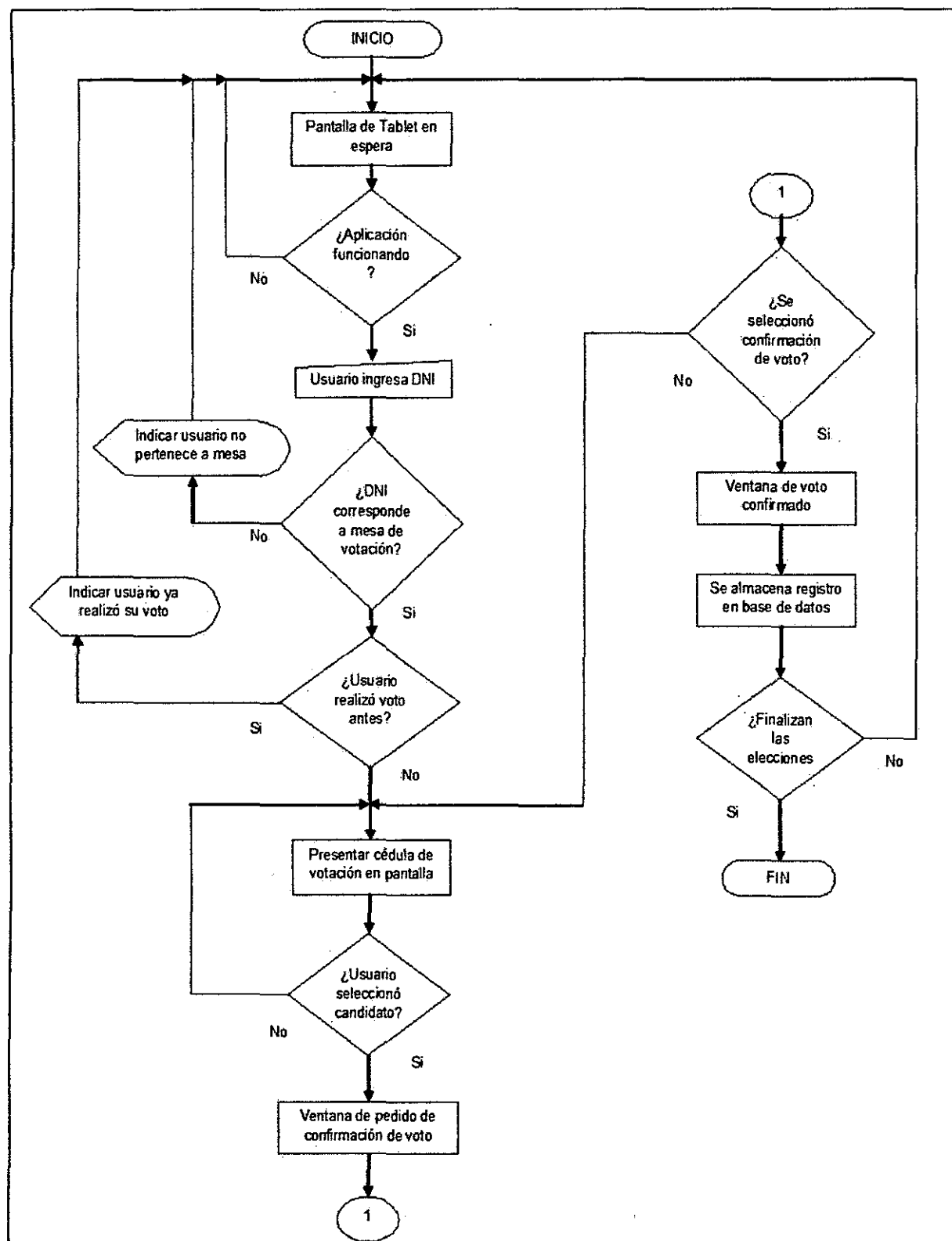
#### **4.3.2. ALGORITMO DE LA APLICACIÓN**

Es importante recalcar el funcionamiento de la aplicación para los procesos electorales electrónicos.

- ✓ La aplicación estará en funcionamiento a la espera del votante.
- ✓ El votante llega a la cabina de votación e ingresa su DNI a través de un teclado numérico.
- ✓ El programa corrobora en su base de datos si ese DNI pertenece a esa mesa de votación, de pertenecer corrobora si ese votante ya realizó su voto o no, y si no ha votado mostrará la cédula de votación donde se presenta la foto y el nombre del candidato.
- ✓ El votante selecciona y confirma su votación.
- ✓ Se almacena los datos de votación en la base de datos de la aplicación.

#### **4.3.3. DIAGRAMA DE FLUJO DE LA APLICACIÓN**

En la Figura 15 se muestra el diagrama de flujo del funcionamiento de la primera parte de la aplicación, es decir la parte que corresponde al sistema de votación electrónica en sí.



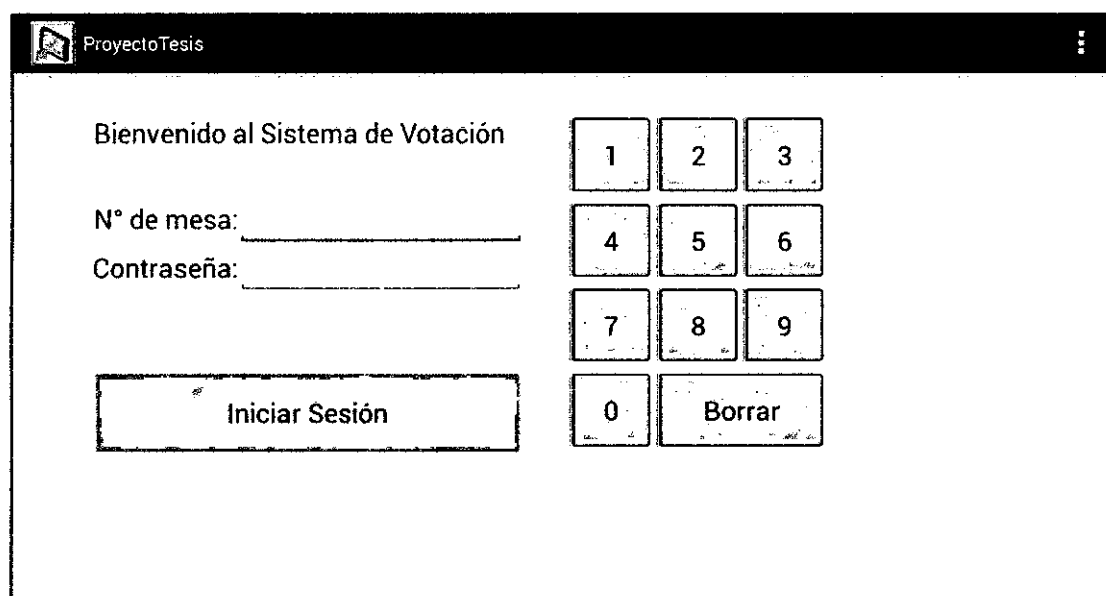
**Figura 15** Diagrama de flujo de aplicación de voto electrónico

#### 4.3.4. DISEÑO DE LA APLICACIÓN

Como se mencionó antes, la aplicación permite que se realice la votación a través de la pantalla de una tableta Android, donde se será capaz de hacer la votación, almacenar los resultados y enviarlos mediante Bluetooth a un hardware diseñado con microcontrolador. A su vez, la aplicación debe poder identificar al votante y alertar si es que el votante pertenece a la mesa de votación y si es que el votante ya ha realizado el escrutinio antes.

En este aspecto, al realizar la aplicación en el software Eclipse, se pensó en contar con cuatro etapas y/o ventanas principales a mostrarse en la pantalla del dispositivo (Tableta).

En la primera etapa (ver Figura 16) se presenta una interfaz (llamada actividad en Android) donde se dará la opción al(los) miembro(s) de mesa para que active la aplicación a través del ingreso de la mesa de votación y la clave de ingreso. A esta ventana o interfaz se le llamará *configuración de mesa* (activity\_main.xml en eclipse).



ProyectoTesis

Bienvenido al Sistema de Votación

N° de mesa: \_\_\_\_\_

Contraseña: \_\_\_\_\_

Iniciar Sesión

1 2 3

4 5 6

7 8 9

0 Borrar

**Figura 16** Primera ventana de la aplicación, actividad llamada “configuración de mesa”

En este punto, se debe tener en cuenta que se está trabajando con bases de datos en SQLite [24], que es un sistema de gestión de bases de datos relacional compatible

con ACID (Atomicity, Consistency, Isolation and Durability), contenida en una relativamente pequeña (~275 KB) biblioteca escrita en C. De esta manera se ha creado una base de datos que contendría el identificador de las mesas de votación y su respectiva contraseña (Ver Figura 17).

Database Structure		Browse Data	Execute SQL
Table: <b>numeroMesa</b>		New Record	Delete Record
mesa	contrasena		
1	014050	0501	
2	014051	0511	
3	014052	0521	
4	014053	0531	
5	014054	0541	

**Figura 17** Ejemplo de base de datos para mesas de votación “mesas.db”

Para acceder a las bases de datos desde el programa en Eclipse se ha creado un archivo .java especial para cada base de datos a trabajar en este proyecto, para este caso se ha llamado *CopiarBaseMesas.java* y la base de datos a importar se llama *mesas.db*. La estructura de cada archivo .java es igual uno del otro y solo difieren en el llamado a la base de datos. Dicha estructura tiene el siguiente algoritmo:

- ✓ SQLiteOpenHelper. Crea, actualiza y conecta con una base de datos SQLite y posee dos métodos, onCreate() para crear tablas si no existen y onUpgrade() para actualizar tablas si es necesario.
- ✓ Se crean los métodos para crear, verificar, y/o copiar una base de datos.
- ✓ Se crea una base de datos en blanco en la dirección "/data/data/com.hassan.proyectotesis/databases/", seguido se busca la base de datos propiamente dicha que se encuentra dentro de la memoria (base de datos que contendría las mesas y claves), para finalmente copiar esa base de datos a la carpeta por defecto de la aplicación para poder trabajarla en conjunto con la app.

En la Figura 18 y 19 se puede apreciar un trozo de código del programa en java para incluir las bases de datos originales para trabajarlas dentro de la aplicación.

```

public class CopiarBaseMesas extends SQLiteOpenHelper {

    //Ruta por defecto de las bases de datos y variables
    private static String DB_PATH = "/data/data/com.hassan.proyectotesis/databases/";
    private static String DB_NAME = "mesas.db";
    private SQLiteDatabase myBaseMesas;
    private final Context myContext;

    // Constructor
    public CopiarBaseMesas(Context context) {
        super(context, DB_NAME, null, 1);
        this.myContext = context;
    }

    //Crea una base de datos vacía en el sistema y la reescribe con nuestro fichero de base de datos.
    public void createDataBase() throws IOException{
        boolean dbExist = checkDataBase();
        if(dbExist){
            //La base de datos existe y no hacemos nada.
        }else{
            //Se crea la base de datos vacía en la ruta por defecto del sistema de nuestra aplicación
            this.getReadableDatabase();
            try {
                copyDataBase();
            } catch (IOException e) {
                throw new Error("Error copiando Base de Datos");
            }
        }
    }
}

```

Figura 18 Parte del código para manejar base de datos, método de creación

```

//Comprueba si la base de datos existe para evitar copiar siempre el fichero cada vez que se
private boolean checkDataBase(){
    SQLiteDatabase checkDB = null;
    try{
        String myPath = DB_PATH + DB_NAME;
        checkDB = SQLiteDatabase.openDatabase(myPath, null, SQLiteDatabase.OPEN_READONLY);
    }catch(SQLiteException e){
        //si llegamos aquí es porque la base de datos no existe todavía.
    }
    if(checkDB != null){
        checkDB.close();
    }
    return checkDB != null ? true : false;
}

//Copia nuestra base de datos desde la carpeta assets a la recién creada
private void copyDataBase() throws IOException{
    //Abrimos el fichero de base de datos como entrada
    InputStream myInput = myContext.getAssets().open(DB_NAME);
    //Ruta a la base de datos vacía recién creada
    String outFileName = DB_PATH + DB_NAME;
    //Abrimos la base de datos vacía como salida
    OutputStream myOutput = new FileOutputStream(outFileName);
    //Transferimos los bytes desde el fichero de entrada al de salida
    byte[] buffer = new byte[1024];
    int length;
    while ((length = myInput.read(buffer))>0){
        myOutput.write(buffer, 0, length);
    }
}

```

Figura 19 Parte del código para manejar base de datos, verificación y copia

Con respecto a la parte del código de la actividad *configuración de mesa* se aprecia en la Figura 16 que contiene 11 elementos button para el teclado de ingreso de los números de mesa de votación y la contraseña. Una vez que la base de datos original



está disponible, cumple la función de permitir la habilitación de la mesa de votación. Cada miembro de mesa debe conocer su número de mesa y contraseña e ingresarla en la pantalla e iniciar sesión. Para ello se hace el programa principal de dicha actividad llamado *MainActivity.java*, donde se hace el llamado a la base de datos con la sentencia *CopiarBaseMesas MDB = new CopiarBaseMesas(getApplicationContext());* y donde se hace una lectura de cada fila y columna de dicha base a través de los comandos *query()* y *openDatabase()*. En la Figura 20 se muestra parte del código para manejar las filas y columnas de la base de datos y compararlas con el texto ingresado en los *TextFields* de la actividad. Y en la Figura 21 se muestra el código base de los botones.

```
public void sesion(View view){
    String textet1 = et1.getText().toString();
    String textet2 = et2.getText().toString();
    String DB_PATH = "/data/data/com.hassan.proyectotesis/databases/";
    String DB_NAME = "mesas.db";
    String myPath = DB_PATH + DB_NAME;
    SQLiteDatabase myBaseMesas = SQLiteDatabase.openDatabase(myPath, null, SQLiteDatabase.OPEN_READONLY);
    Cursor c = myBaseMesas.query("numeroMesa", new String[] {"mesa", "contrasena"}, null, null, null, null,
    c.moveToFirst();
    while (c.isAfterLast() == false) {
        textot1 = c.getString(0).toString();
        textot2 = c.getString(1).toString();
        if ((textot1.equals(textet1)) && (textot2.equals(textet2))){
            break;
        }else{
            c.moveToNext();
        }
    }
    if ((textot1.equals(textet1)) && (textot2.equals(textet2))){
        Intent i = new Intent(this, FirstActivity.class );
        i.putExtra("mesaPertenece", textet1);
        i.putExtra("contrasenaMesa", textet2);
        startActivity(i);
        myBaseMesas.close();
        finish();
    }else{
        Toast toast = Toast.makeText(this, "Usuario o contraseña inválida", Toast.LENGTH_LONG);
        toast.setGravity(Gravity.CENTER_VERTICAL, 0, 0);
        toast.show();
        et1.setText("");
        et2.setText("");
        et1.requestFocus();
    }
}
```

**Figura 20** Parte del código que maneja la base en la actividad “configuración de mesa”

Como es de esperarse el código lee la base de datos en donde hace un barrido con el comando *moveToNext()* hasta encontrar el número de mesa y corroborar la contraseña. De no ser así mostrará un mensaje de “Usuario o contraseña inválida”. Cuando encuentre la mesa y compruebe la contraseña de dicha mesa, hará el llamado a la siguiente actividad con el comando *startActivity()* para dirigirse a la pantalla de ingreso del DNI.

```
//Método para el botón 8
public void numeroOcho(View view){
    if (et1.isFocused() == true){
        texto = et1.getText().toString() + "8";
        et1.setText(texto);
    }
    if (et2.isFocused() == true){
        texto = et2.getText().toString() + "8";
        et2.setText(texto);
    }
}
```

Figura 21 Codificación para los botones numéricos

Una vez que se haya ingresado correctamente, se presenta en pantalla la segunda actividad o interfaz que será la primera de la aplicación de voto propiamente dicha. Esta actividad llamada *activity\_first.xml* y para efectos de comodidad le diremos *ingreso de DNI*, tendrá también un teclado numérico y un TextField. Tal actividad puede observarse en la Figura 22.

Figura 22 Actividad de ingreso de DNI

Es esta instancia la aplicación debe:

- ✓ Leer el texto ingresado como DNI.
- ✓ Identificar que el número ingresado sea de 8 dígitos y sino arrojar un mensaje con la advertencia.

- ✓ Al momento de seleccionar en Inicie Votación debe conectarse a la base de datos y verificar que el DNI pertenece a la mesa de votación configurada y que el DNI no ha votado antes.
- ✓ Finalmente enviar a la siguiente actividad donde se presentará la cédula de votación.
- ✓ También debe poseer un menú de configuración para que el encargado o miembro de mesa pueda cerrar la sesión de votación en cualquier momento, en caso de alguna emergencia.

Para cumplir con las especificaciones mencionadas, nuevamente se hace uso de bases de datos, creando un archivo .java para la importación de la base de datos llamada *votantes.db* que contendrá los datos de DNI, apellidos, nombres y mesa a la que pertenece la persona (Ver Figura 23).

Database Structure

Browse Data

Execute SQL

Table: 

datosVotantes

New Record

Delete Record

	mesa	apellidos	nombres	dni	
1	014050	KHALIL CASTRO	IRIS WASILA	02011001	
2	014050	KHALIL CASTRO	ROSA DE FATIMA	02011002	
3	014050	KHALIL CASTRO	SORAYA DEL ROCIO	02011003	
4	014050	KHALIL CASTRO	NAHIDA	02011004	
5	014050	CASTRO SEMINARIO	YOLANDA IRIS	02011005	
6	014050	KHALIL ABDEL	HASSAN	02011006	
7	014050	DELGADO KHALIL	HASSAN	02011007	
8	014050	FLORES KHALIL	ANDREA DE FATIMA	02011008	
9	014050	URBINA KHALIL	SHADIA SOFIA	02011009	
10	014050	URBINA KHALIL	ARIANA SOFIA	02011010	
11	014051	DUQA KHALIL	ALIA NAEL	02012011	
12	014051	DUQA KHALIL	HADEEL NAEL	02012012	
13	014051	DELGADO ACUÑA	GUILTERMO	02012013	
14	014051	DELGADO ACUÑA	SANDRA	02012014	
15	014051	DELGADO ACUÑA	BERTILA	02012015	
16	014051	DELGADO ACUÑA	MERLY	02012016	
17	014051	DELGADO ACUÑA	MERCEDES	02012017	
18	014051	DELGADO ACUÑA	MARCO ANTONIO	02012018	

Figura 23 Ejemplo de base de datos para identificación de DNI “votantes.db”

Similar a la primera parte, el archivo .java creado se llama *CopiarBaseVotantes* que se ejecuta de la misma manera que el archivo del caso anterior, es decir cumple con el algoritmo:

- ✓ SQLiteOpenHelper. Crea, actualiza y conecta con una base de datos SQLite.
- ✓ Se crean los métodos para crear, verificar, y/o copiar una base de datos.
- ✓ Se crea una base de datos en blanco en la dirección "/data/data/com.hassan.proyectotesis/databases/", seguido se busca la base de datos propiamente dicha que se encuentra dentro de la memoria (base de datos que contendría las mesas y claves), para finalmente copiar esa base de datos a la carpeta por defecto de la aplicación para poder trabajarla en conjunto con la app.

En la Figura 24 se aprecia parte del código del archivo *CopiarBaseVotantes*.

```
public class CopiarBaseVotantes extends SQLiteOpenHelper {

    //Ruta por defecto de las bases de datos y variables
    private static String DB_PATH = "/data/data/com.hassan.proyectotesis/databases/";
    private static String DB_NAME = "votantes.db";
    private SQLiteDatabase myBaseVotantes;
    private final Context myContext;

    // Constructor
    public CopiarBaseVotantes(Context context) {
        super(context, DB_NAME, null, 1);
        this.myContext = context;
    }

    //Crea una base de datos vacía en el sistema y la reescribe con nuestro fichero de
    public void createDataBase() throws IOException{
        boolean dbExist = checkDataBase();
        if(dbExist){
            //La base de datos existe y no hacemos nada.
        }else{
            //Se crea la base de datos vacía en la ruta por defecto del sistema de nues
            this.getReadableDatabase();
            try {
                copyDataBase();
            } catch (IOException e) {
                throw new Error("Error copiando Base de Datos");
            }
        }
    }
}
```

Figura 24 Parte del código para importar la base de datos "votantes.db"

Funciona importando la base de datos *votantes.db* a la carpeta de bases de datos por defecto con la que trabaja la aplicación.

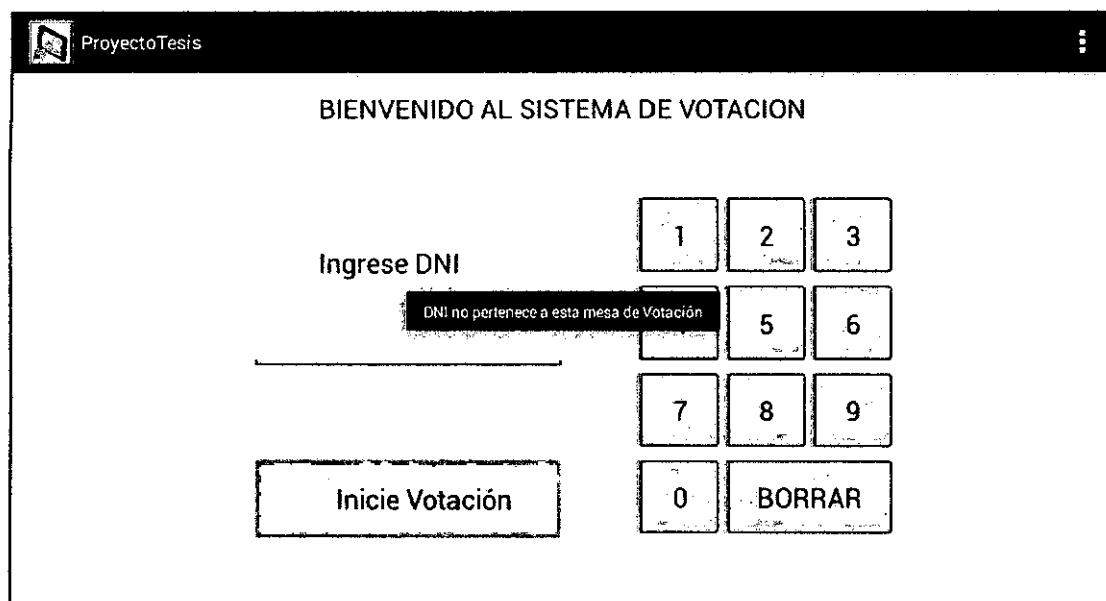
Con respecto a la parte del código de la actividad *ingreso de DNI* se aprecia en la Figura 22 que contiene 11 elementos button para el teclado de ingreso del DNI del votante. Una vez que la base de datos original está disponible, cumple la función de verificar si el votante pertenece a la mesa de votación abierta en sesión y si el votante ha realizado antes el escrutinio o no. Para ello se hace el programa principal de dicha actividad llamado *FirstActivity.java*, donde se hace el llamado a la base de datos con la sentencia `CopiarBaseVotantes MDB = new CopiarBaseVotantes(getApplicationContext());` y donde se hace una lectura de cada fila y columna de dicha base a través de los comandos *query()* y *openDatabase()*. En la Figura 25 se muestra parte del código para manejar las filas y columnas de la base de datos y compararlas con el texto ingresado en el TextField de la actividad.

```
public void sesion2(View view){
    Bundle bundle = getIntent().getExtras();
    String textet11 = et11.getText().toString();
    String textPert = bundle.getString("mesaPertenece");
    String contrasenaPert = bundle.getString("contrasenaMesa");
    String APELLIDOS2 = "";
    String NOMBRES2 = "";
    String APELLIDOS = "";
    String NOMBRES = "";
    String HORAVOTACION = "";
    String DB_PATH = "/data/data/com.hassan.proyectotesis/databases/";
    String DB_NAME1 = "votantes.db";
    String DB_NAME2 = "votacion.db";
    String myPath1 = DB_PATH + DB_NAME1;
    String myPath2 = DB_PATH + DB_NAME2;
    SQLiteDatabase myBaseVotantes = SQLiteDatabase.openDatabase(myPath1, null, SQLiteDatabase.OPEN_READONLY);
    SQLiteDatabase myBaseVotacion = SQLiteDatabase.openDatabase(myPath2, null, SQLiteDatabase.OPEN_READWRITE);
    myBaseVotacion.execSQL("CREATE TABLE IF NOT EXISTS " + "mesa_" + textPert + "(dni TEXT, apellidos TEXT, nombres TEXT, hora TEXT)");
    Cursor c = myBaseVotantes.query("datosVotantes", new String[] {"mesa", "apellidos", "nombres", "dni"}, null, null, null, null, null);
    Cursor d = myBaseVotacion.query("mesa_" + textPert, new String[] {"dni", "apellidos", "nombres", "hora"}, null, null, null, null, null);
    c.moveToFirst();
    d.moveToFirst();
    while (c.isAfterLast() == false) {
        mesaPert = c.getString(0).toString();
        dni1 = c.getString(3).toString();
        APELLIDOS = c.getString(1).toString();
        NOMBRES = c.getString(2).toString();
        if (dni1.equals(textet11)) {
            break;
        }
        else {
            c.moveToNext();
        }
    }
}
```

Figura 25 Parte del código que maneja la base en la actividad “ingreso de DNI”

De la Figura 25 se puede observar que en esta parte del proyecto se trabaja con dos bases de datos. Una es la ya mencionada *votantes.db* que es importada y abierta con el comando *myBaseVotantes* para comprobar los datos del votante y si su DNI pertenece a la mesa de votación configurada al comienzo. La otra es una base de datos que llamaré *votacion.db* que es importada y trabajada por el comando *myBaseVotacion* y en donde se va a registrar solamente el DNI de la persona votante para la

comprobación y constancia del sufragio, aquí se comprobará si el DNI ya había realizado antes el sufragio, y de no ser así, se registrará en dicha base de datos; y si el DNI no pertenece a dicha mesa de votación se avisará por pantalla (Ver Figura 26).



**Figura 26** Aviso de DNI no corresponde a la mesa de Votación

También se ha desarrollado un menú en el cual se puede cambiar de mesa de votación, para hacer independiente la aplicación y la tableta y se pueda configurar cualquier mesa de votación en cualquier lugar (por ejemplo una emergencia o un reemplazo de dispositivo), así como un menú para enviar los datos vía Bluetooth al finalizar la votación (se detallará más adelante), y por último un menú para salir de la aplicación. En la Figura 27 se puede ver las opciones de menú disponibles.

En cualquiera de los casos o selección de un menú se requerirá la contraseña que los miembros de mesa deben saber correspondiente a la mesa de votación que esté funcionando en ese momento (Ver Figura 28). Cabe resaltar que cada vez que se cambie de mesa, la aplicación grabará en la misma base de datos el DNI del votante con su respectiva mesa de votación para la confirmación de sufragio.

ProyectoTesis

BIENVENIDO AL SISTEMA DE VOTACION

Cambiar Mesa  
Enviar Datos  
Salir

Ingrese DNI

1 2 3  
4 5 6  
7 8 9  
0 BORRAR

Inicie Votación

**Figura 27** Menú de Opciones

ProyectoTesis

BIENVENIDO AL SISTEMA DE VOTACION

Cambiar Mesa

Ingrese contraseña de mesa 014050

Confirmar

Inicie Votación

0 BORRAR

**Figura 28** Solicitud de contraseña para uso de menú de opciones

Una vez que se haya ingresado correctamente el DNI y se haya comprobado que pertenece a la mesa de votación establecida y se haya comprobado que el votante no ha realizado el sufragio anteriormente, se presenta en pantalla la tercera actividad o interfaz que será la segunda de la aplicación de voto propiamente dicha. Esta actividad llamada *activity\_second.xml* y a la cual le diremos *selección de candidato*, tendrá un esquema muy parecido a las cédulas de votación de papel en donde se presentará el nombre y las fotos de los candidatos y donde el votante podrá seleccionar y hacer su votación con

solo tocar en la pantalla sobre el nombre o foto. Aparecerá entonces una ventana de confirmación para que el elector confirme su votación. En este paso ya no se podrá regresar y el elector debe hacer su votación. En la Figura 29 y 30 se muestra la interfaz de la cédula de votación.

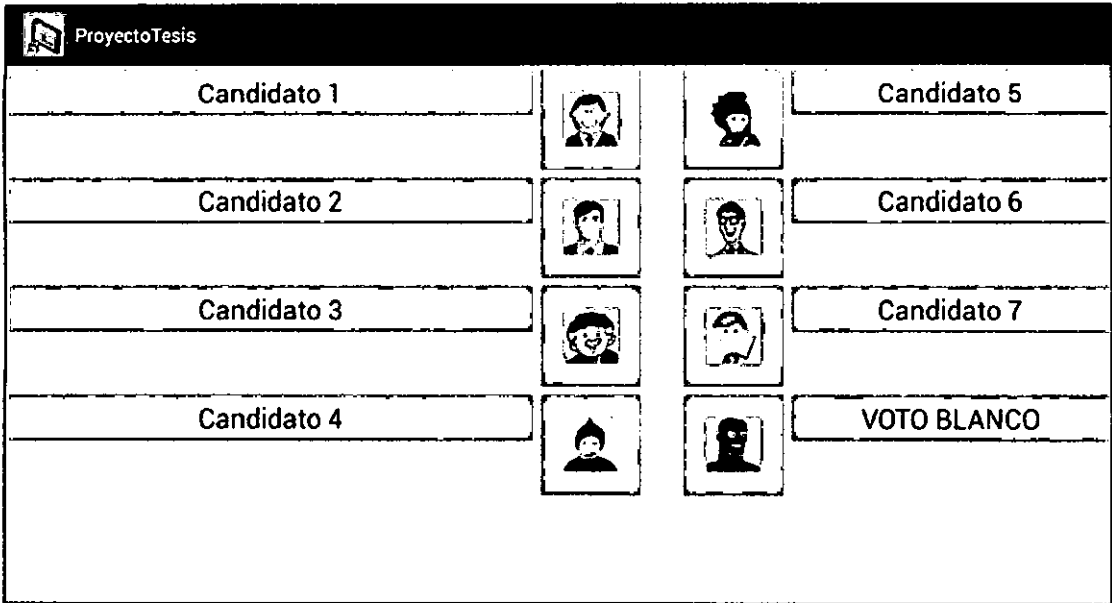


Figura 29 Cédula de votación

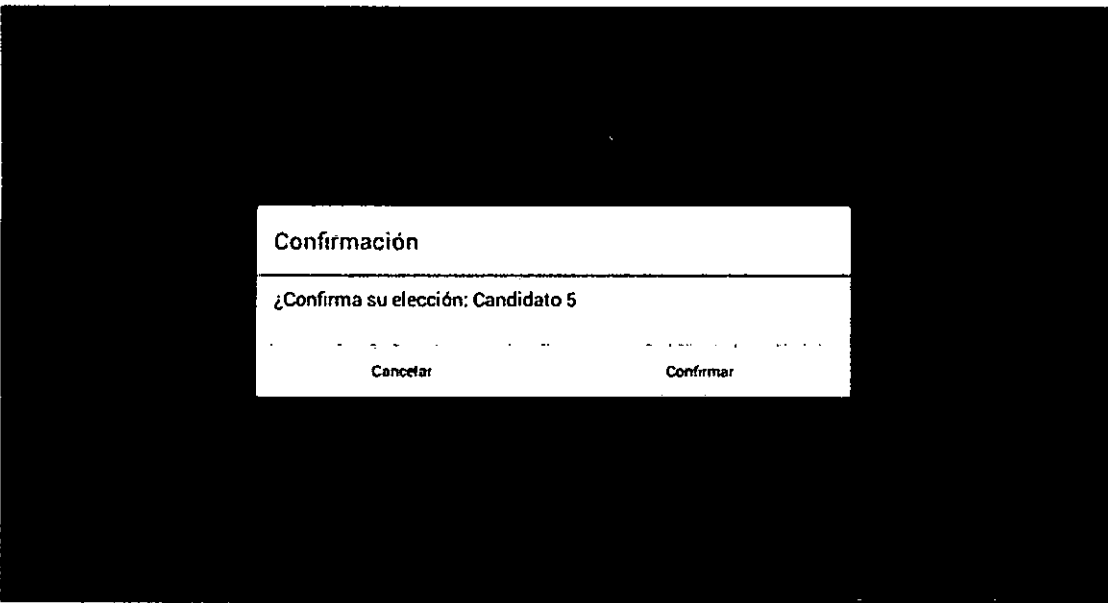
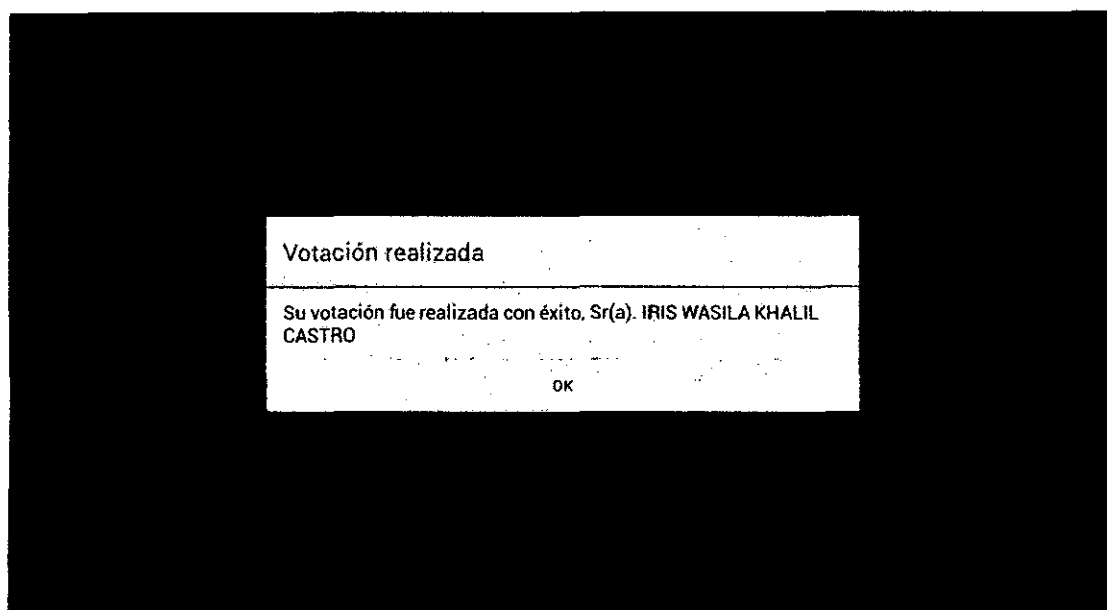


Figura 30 Confirmación de votación



Una vez seleccionada la confirmación la aplicación muestra una ventana de confirmación y regresará a la pantalla de *ingreso de DNI* (Ver Figura 31). La aplicación también presenta la opción de voto en blanco como se aprecia en la Figura 29.



**Figura 31** Voto realizado

En esta actividad también se ha trabajado con bases de datos. En este caso se trabaja con una base de datos que se llama *conteo.db*. En ella se almacena el nombre del candidato (o un número que le represente) y también la cantidad de votos que va recibiendo a través de un contador en el código, pero no se almacena el nombre ni datos de los electores. En la Figura 32 se presenta una parte del código que realiza esto.

Todos los datos se han ido transmitiendo de actividad en actividad, es decir, el número de mesa, el número de DNI, el nombre y apellidos del elector. Todo esto es así con la finalidad de poder registrar en cada base de datos y en cada proceso a las personas que realizan el voto y la mesa de votación en la que se trabaja, así como llevar el control de la cantidad de veces que se selecciona a un candidato y hacer el conteo del escrutinio. También esto ayuda a poder tener acceso o no a ciertas partes de la aplicación como el menú de opciones al que solo pueden acceder el(los) miembro(s) de mesa a través de la clave de acceso.


```

//Actualización de votación
Cursor c = myBaseConteo.query("mesa_" + mesaPertenece, new String[] {"id", "candidato", "votos"}, null, null);
c.moveToFirst();
while (c.isAfterLast() == false) {
    myid = c.getString(0).toString();
    conteo = c.getString(2).toString();
    if (myid.equals(representacion)) {
        int contador = Integer.parseInt(conteo);
        contador = contador + 1;
        String valorConteo = Integer.toString(contador);
        ContentValues voto = new ContentValues();
        voto.put("id", myid);
        voto.put("votos", valorConteo);
        myBaseConteo.update("mesa_" + mesaPertenece, voto, "id=" + myid, null);
        myBaseConteo.close();
        break;
    } else {
        c.moveToNext();
    }
}
c.close();

```

**Figura 32** Parte del código para trabajar base de datos para llevar conteo de escrutinio

Cada vez que se realiza un voto, la aplicación actualiza la base de datos con el comando *myBaseConteo.update()* en donde se añade al valor actual la suma de valor uno y se reescribe en la columna llamada votos. En la Figura 33 se muestra el resultado almacenado tras cinco ejemplos de votaciones en una mesa de votación, el cual corresponde a la base de datos *conteo.db*.

Table: <input type="text" value="mesa_014050"/> 		
id	candidato	votos
1	1 Candidato 1	0
2	2 Candidato 2	0
3	3 Candidato 3	0
4	4 Candidato 4	0
5	5 Candidato 5	1
6	6 Candidato 6	0
7	7 Candidato 7	0
8	8 Candidato 2	5

**Figura 33** Ejemplo de votos contabilizados en base de datos

Se observa que se ha registrado en la base de datos una tabla con el nombre de la mesa de votación y donde se ve en cada fila el nombre del candidato y las veces que ha sido seleccionado durante las votaciones. Cada vez que en un dispositivo se cambie la mesa de votación se creará una nueva tabla con el nombre de la mesa y con el conteo de los votos correspondientes a cada candidato.

## 4.4. APLICACIÓN DE ENVÍO DE DATOS

Como se mencionó en el apartado 4.3.1 y se mostró en la Figura 14 en el diagrama de bloques de la aplicación y en la Figura 27 menú de opciones, en la aplicación de votación se diseña una parte para enviar los datos del escrutinio una vez finalizado el proceso electoral a través del módulo Bluetooth. Con esto se logrará transmitir los resultados del proceso electoral a un hardware con microcontrolador (que será explicado en el siguiente capítulo). Esto sería beneficioso para el traslado de la información y de la aplicación a los lugares más difíciles de acceder en nuestro país como se mencionó en los capítulos introductorios.

### 4.4.1. DIAGRAMA DE FLUJO

En la Figura 34 se muestra el diagrama de flujo del funcionamiento de esta parte de la aplicación, es decir, de la selección de *Enviar Datos* por Bluetooth.

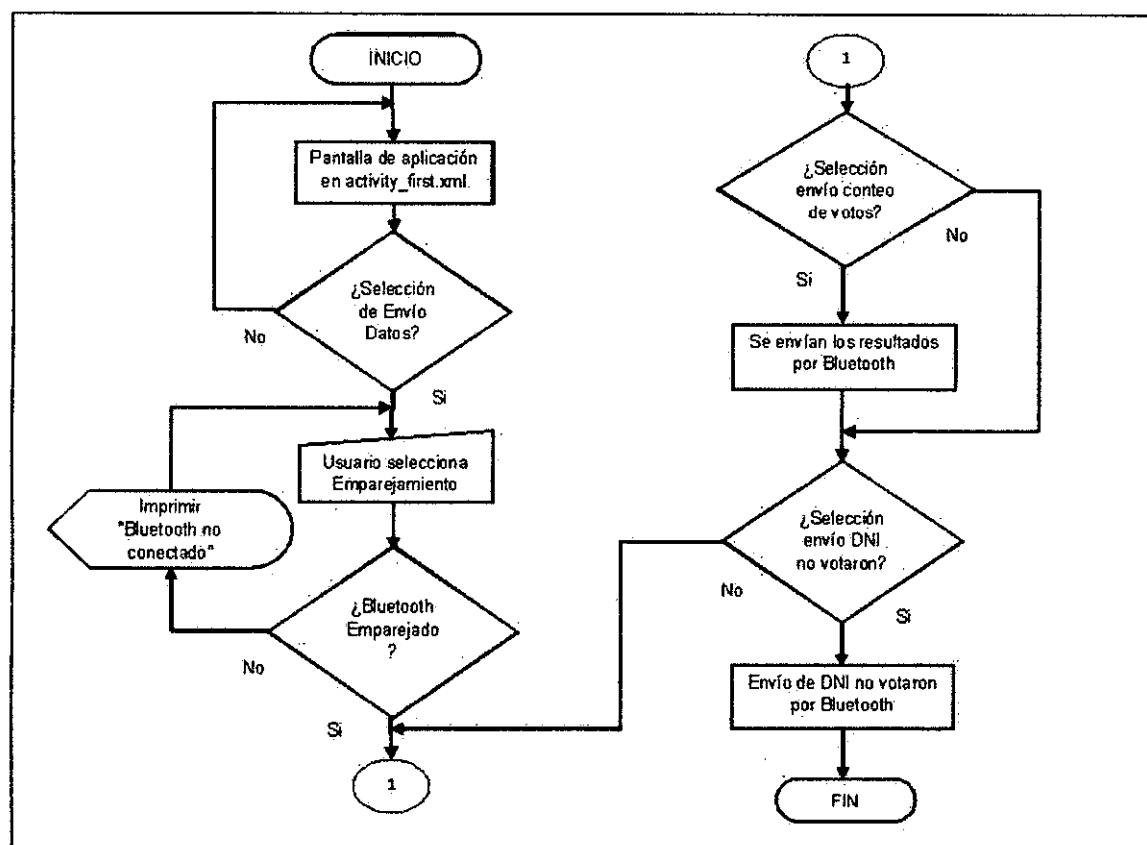
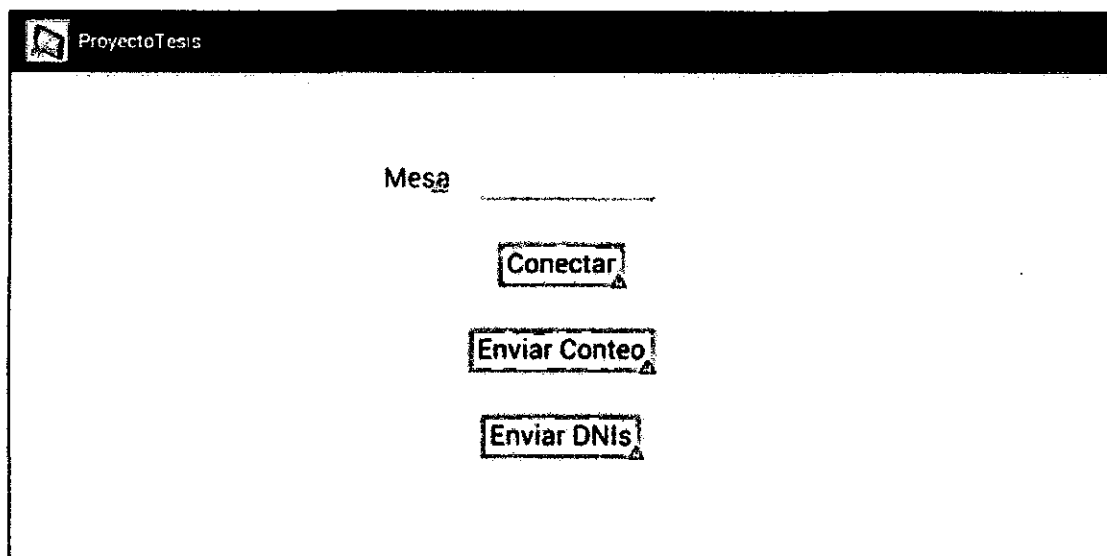


Figura 34 Diagrama de flujo de envío de datos

#### 4.4.2. ACTIVIDAD DE LA APLICACIÓN

Como se vio en la Figura 27 (menú de opciones), se ha establecido la opción de *Enviar Datos* la cual pertenece a la actividad *ingreso de DNI (activity\_first.xml)*, para mayor comodidad de uso de la aplicación, conducirá a una nueva actividad a la cual llamaré *envío de datos* (en eclipse se llama *activity\_third.xml*).



**Figura 35** Actividad de envío de datos

En la Figura 35 se puede observar que la interfaz del envío de datos posee tres buttons, el primero llamado *Conectar* para emparejar el Bluetooth de la tableta con el hardware de microcontrolador, y los otros dos para la transmisión de los datos en sí. El segundo llamado *Enviar Conteo* enviará los resultados del escrutinio con la cantidad de votos perteneciente a cada candidato, mientras el tercero llamado *Enviar DNIs* será para que el encargado envíe los DNIs de las personas que no realizaron su voto durante el periodo establecido. Es destacable que este proceso será posible solamente para cada mesa de votación respectivamente.

Cuando la actividad inicia, se hace la carga de dos bases de datos a través de los llamados *baseVotos()* y *baseDni()*. El primero abre la base de datos de los resultados del escrutinio para almacenar en variables los resultados. El segundo abre la base de datos donde se almacenan los DNIs de las personas que no votaron para su posterior envío.

En la Figura 36 se puede apreciar parte del código de conexión Bluetooth diseñada para el envío de datos. En esta parte, se han creado tres métodos llamados *conectar()*, *envioUno()* y *envioDos()*, los cuales responden al presionar cada button mencionado en el párrafo anterior, respectivamente. El primero, *conectar()* responde al llamado del button *Conectar* en donde se obtiene el dispositivo bluetooth con el código *BluetoothAdapter.getDefaultAdapter()* y *BluetoothDevice*, y se crea la variable *ntBTsocket* como el hilo de conexión entre la aplicación y el dispositivo HC-06. Cuando estén conectados, estamos listos para enviar los datos. Con presionar el button *Enviar Conteo* se llama al método *envioUno()* que enviará como símbolo de identificación el "\$" seguido de un "0" y luego los dígitos de la mesa de votación correspondiente (el "0" será para que el microcontrolador sepa que subrutina seguir, se explicará en el siguiente capítulo). Después enviará el identificador "&" seguido de los resultados del conteo por cada candidato en un formato de 3 dígitos. Con presionar el button *Enviar DNIs* se llama al método *envioDos()* que enviará como identificador el "\$" seguido con un "1" y los dígitos de la mesa de votación correspondiente (el "1" será para que el microcontrolador sepa que subrutina seguir, se explicará en el siguiente capítulo). Después hará un barrido por cada fila de la base de datos *novoto.db* (donde se almacenan los DNIs de los que no hayan votado) para enviar la cadena "%" + "4 primeros dígitos del DNI" + "4 segundos dígitos del DNI" + "valor 0 o 1" a través de bluetooth. Cuando llegue a la última fila enviará como último valor el "0" con lo que el microcontrolador sabrá que será el último dato a recibir y escribir en el datalogger.

```
//Método para enviar conteo
public void conectar(View view) {
    bt31 = (Button)findViewById(R.id.bt31);
    if (bt31.getText().equals("Conectar")){
        Toast toast;
        try {
            // Obtenemos el adaptador bluetooth
            BluetoothAdapter btAdapter = BluetoothAdapter.getDefaultAdapter();
            // Recorremos la lista de dispositivos descubiertos
            Set<BluetoothDevice> bondedDevices = btAdapter.getBondedDevices();
            BluetoothDevice ntDevice = null;
            for (BluetoothDevice bluetoothDevice : bondedDevices) {
                if (bluetoothDevice.getName().equals(miNT.toString())) {
                    ntDevice = bluetoothDevice;
                    break;
                }
            }
            ntBTsocket = ntDevice.createRfcommSocketToServiceRecord(UUID.fromString("00001101-0000-1000-8000-00805F9B34FB"));
            ntBTsocket.connect();
        } catch (IOException ioe) {
            // Error en la escritura
        }
        bt31.setText("Desconectar");
    }
}
```

Figura 36a Parte del código para conectar con HC-06

```

//Enviar resultados de escrutinio por BT
private void envioUno(){
    try{
        ntDos = new DataOutputStream(ntBTsocket.getOutputStream());
        Bundle bundle = getIntent().getExtras();
        String mesaPertenece = bundle.getString("mesaPertenece");
        String mes1 = mesaPertenece.substring(0, 1);
        String mes2 = mesaPertenece.substring(1, 2);
        String mes3 = mesaPertenece.substring(2, 3);
        String mes4 = mesaPertenece.substring(3, 4);
        String mes5 = mesaPertenece.substring(4, 5);
        String mes6 = mesaPertenece.substring(5, 6);
        //Envia valor de espera "0"
        ntDos.write(new String("$").getBytes());
        ntDos.write(new String("0").getBytes());
        ntDos.write(mes1.getBytes());
        ntDos.write(mes2.getBytes());
        ntDos.write(mes3.getBytes());
        ntDos.write(mes4.getBytes());
        ntDos.write(mes5.getBytes());
        ntDos.write(mes6.getBytes());
        for(int q=0;q<500000000;q++){
            //tiempo de espera
        }
        //Enviar datos
        ntDos.write(new String("&").getBytes());
        ntDos.write(cont1.getBytes());
        ntDos.write(cont2.getBytes());
        ntDos.write(cont3.getBytes());
        ntDos.write(cont4.getBytes());
        ntDos.write(cont5.getBytes());
        ntDos.write(cont6.getBytes());
        ntDos.write(cont7.getBytes());
        ntDos.write(cont8.getBytes());
    } catch (IOException ioe) {
        // Error en la escritura
    }
}

```

Figura 36b Parte del código para enviar resultados de escrutinio a HC-06

```

String DB_PATH = "/data/data/com.hassan.proyectotesis/databases/";
String DB_NAME = "novoto.db";
String myPath = DB_PATH + DB_NAME;
SQLiteDatabase myBaseNovoto = SQLiteDatabase.openDatabase(myPath, null, SQLiteDatabase.OPEN_READWRITE);
Cursor e = myBaseNovoto.query("mesa_" + mesaPertenece, new String[] {"dni", "apellidos", "nombres"}, null,
e.moveToFirst();
while (e.isAfterLast() == false) {
    for(int q=0;q<500000000;q++){
        //tiempo de espera
    }
    if (e.isLast() == true){
        confirma = "0";
        ntDos.write(new String("%").getBytes());
        ntDos.write(e.getString(0).toString().substring(0, 4).getBytes());
        ntDos.write(e.getString(0).toString().substring(4, 8).getBytes());
        ntDos.write(confirma.getBytes());
        break;
    }else{
        confirma = "1";
        ntDos.write(new String("%").getBytes());
        ntDos.write(e.getString(0).toString().substring(0, 4).getBytes());
        ntDos.write(e.getString(0).toString().substring(4, 8).getBytes());
        ntDos.write(confirma.getBytes());
        e.moveToNext();
    }
}
}

```

Figura 36c Parte del código para enviar DNIs que no votaron a HC-06

## 4.5. RESUMEN

La aplicación de voto electrónico usando Eclipse para desarrollar el código en Java para Android y la aplicación para enviar los datos resultantes del escrutinio a través de Bluetooth han sido desarrolladas. Estas aplicaciones promueven el uso de dispositivos táctiles que reemplazarían a máquinas más sofisticadas pero de dimensiones y peso sumamente mayor. En este capítulo:

- ✓ Se ha desarrollado una aplicación en Java para Android que presenta tres interfaces o actividades. En la primera interfaz se muestra una pantalla para ingresar el número de mesa de votación y una clave de ingreso que solo debe ser conocida por el encargado y/o el(los) miembro(s) de mesa. Cuando se haya comprobado el número de mesa y su contraseña en la base de datos, se pasará a la segunda interfaz. En ella, se presenta un diseño donde el votante podrá ingresar su DNI a través de un teclado numérico y se ha elaborado un menú de opciones para que el miembro de mesa pueda, con la respectiva contraseña, cambiar de mesa de votación, salir de la aplicación, y enviar los datos del escrutinio por Bluetooth. Al mismo tiempo en esta segunda interfaz se comprobará que el DNI del elector no esté registrado como si ya hubiese realizado su voto y se comprobará en la respectiva base de datos que tal DNI pertenezca a la mesa de votación en funcionamiento. Finalmente, una vez hechas las comprobaciones se pasará a una tercera actividad donde el elector podrá ver la cédula de votación por la pantalla y seleccionar al candidato de su preferencia. La selección quedará registrada en una base de datos donde se guarda el conteo de votos por candidato. El DNI se registra en otra base de datos para que quede constancia de que el elector realizó la votación.
- ✓ También se ha desarrollado como parte de la aplicación una interfaz para que a través del módulo Bluetooth de cualquier dispositivo Android se pueda enviar los resultados del escrutinio por candidato para su futuro procesado junto a los resultados de otras mesas de votación. Se enviarán tanto los resultados de la votación como la base de datos de los DNIs que no hayan efectuado la votación en el tiempo establecido.

## CAPÍTULO V

### DISEÑO DE LA APLICACIÓN – HARDWARE

#### 5.1. COMPONENTES PARA EL DISEÑO

##### 5.1.1. MICROCONTROLADOR PIC

Los microcontroladores PIC (Peripheral interface Controller) [25], son fabricados por la empresa MICROCHIP Technology INC. cuya central se encuentra en Chandler, Arizona, esta empresa ocupa el primer lugar en venta de microcontroladores de 8 bits desde el año 2002. Su gran éxito se debe a la gran variedad (más de 180 modelos), gran versatilidad, gran velocidad, bajo costo, bajo consumo de potencia, y gran disponibilidad de herramientas para su programación. Uno de los microcontroladores más populares en la actualidad es el PIC16F877a (4 MHz) y PIC16F877 (20 MHz).

CARACTERÍSTICAS	16F877
Frecuencia máxima	DX-20MHz
Memoria de programa flash palabra de 14 bits	8KB
Posiciones RAM de datos	368
Posiciones EEPROM de datos	256
Puertos E/S	A,B,C,D,E
Número de pines	40
Interrupciones	14
Timers	3
Módulos CCP	2
Comunicaciones Serie	MSSP, USART
Comunicaciones paralelo	PSP
Líneas de entrada de CAD de 10 bits	8
Juego de instrucciones	35 Instrucciones
Longitud de la instrucción	14 bits
Arquitectura	Harvard
CPU	Risc
Canales Pwm	2
Pila Harware	-
Ejecución En 1 Ciclo Máquina	-

Figura 37 Cuadro de características del PIC16F877 [26]

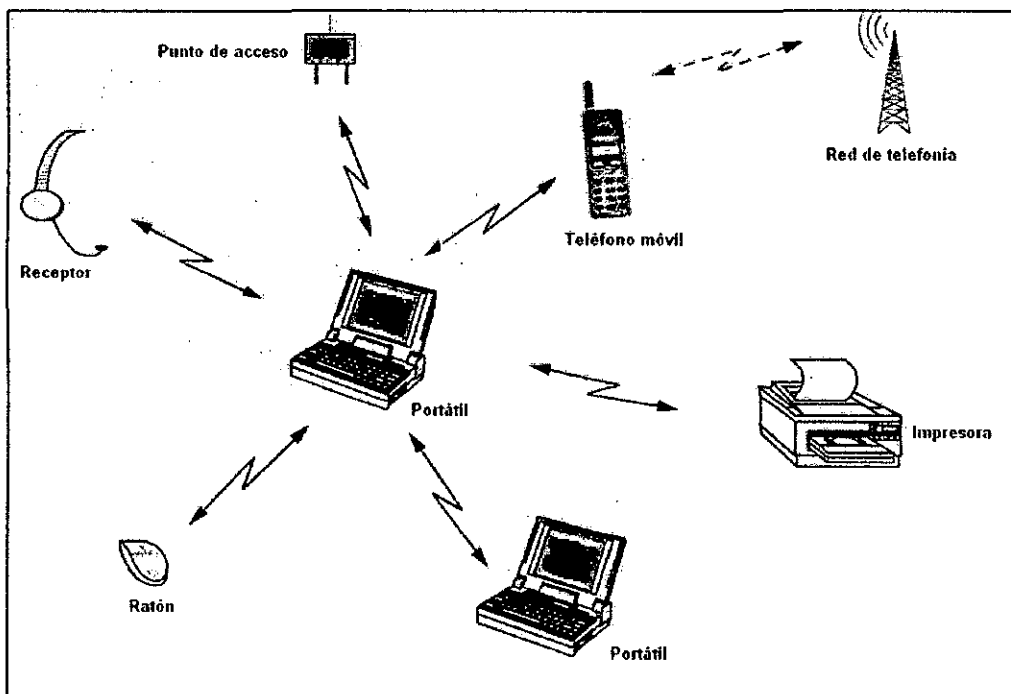


### 5.1.2. BLUETOOTH

El Bluetooth [27] es una tecnología orientada a la conectividad inalámbrica entre dispositivos tan dispares como PCs, PDAs, teléfonos móviles, electrodomésticos, etc. En general, podemos decir que las posibilidades pueden considerarse infinitas. Aparte de ser una nueva tecnología, es también una especificación abierta para comunicaciones inalámbricas de voz y datos. Está basado en un enlace de radio de bajo costo y corto alcance, el cual proporciona conexiones instantáneas (ad hoc, ver Figura 38) tanto para entornos de comunicaciones móviles como estáticos.

El Bluetooth, al ser un estándar abierto, pretende conectar una amplia gama de dispositivo sin importar su marca. Sus principales características son:

- ✓ Robustez
- ✓ Bajo costo
- ✓ Necesidad de poca potencia
- ✓ Baja complejidad
- ✓ Es un estándar global



**Figura 38** Alcance de las especificaciones Bluetooth [27]

### 5.1.3. DATALOGGER

Un registrador de datos (datalogger) [28] es un dispositivo electrónico que registra datos en el tiempo o en relación a la ubicación por medio de instrumentos y sensores propios o conectados externamente. Casi todos están basados en microcontroladores. Por lo general son pequeños, con pilas, portátiles, y equipados con un microprocesador, memoria interna para almacenamiento de datos y sensores. Algunos registradores de datos se comunican con un ordenador personal y utilizan software específico para activar el registrador de datos, ver y analizar los datos recogidos, mientras que otros tienen un dispositivo de interfaz local (teclado, pantalla LCD) y puede ser utilizado como un dispositivo independiente.

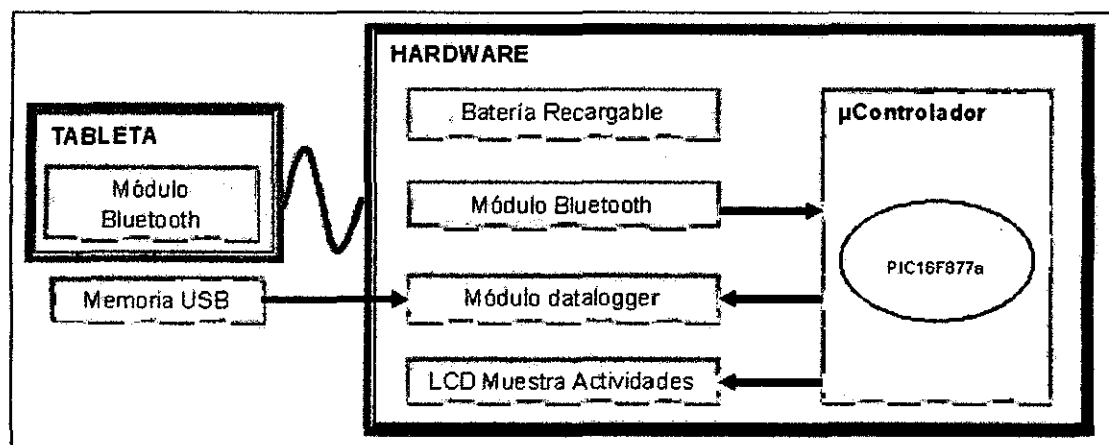
### 5.1.4. OTROS COMPONENTES ELECTRÓNICOS

Para el diseño del hardware también se utilizarán componentes como LCD, resistores, capacitores, cristal de cuarzo, dipswitches o interruptores, reguladores de tensión, etc.

## 5.2. ANÁLISIS DEL PROYECTO

En este capítulo, se explica el diseño propuesto del hardware, el cual es basado en un microcontrolador PIC 16F877a que tendrá conectado a sus puertos principalmente un módulo Bluetooth y un módulo Datalogger, por lo que se divide esta sección en dos partes principales de diseño. Una de ellas es la parte de *adquisición de datos*, donde se ha conectado al PIC un módulo Bluetooth que cumple la función de recepción de los datos enviados desde la aplicación software tal como se mencionó en el capítulo anterior. La segunda parte trata del *almacenamiento* de los datos, el módulo Bluetooth envía estos datos al PIC, y aquí el PIC estará conectado a un módulo Datalogger que es capaz de almacenar los datos en una memoria USB en formato .txt. Los datos serán almacenados en tipo texto parecido a una tabla dentro de una carpeta con el nombre de la mesa de votación para así poder recoger datos de diversas mesas. Finalmente cada proceso realizado ya sea en adquisición de datos o almacenamiento de los mismos podrá ser controlado por la persona encargada de utilizar el hardware mediante un LCD conectado al microcontrolador, donde se indica cada actividad que se esté realizando.

En la Figura 39 se puede apreciar el diagrama de bloques del funcionamiento del diseño propuesto.



**Figura 39** Diagrama de bloques del diseño hardware propuesto

### 5.3. ALGORITMO

Es importante recalcar el funcionamiento del diseño para la adquisición de datos y el almacenamiento de los mismos.

- ✓ El hardware estará apagado hasta que se vaya a hacer el envío de la información, es decir, al finalizar el escrutinio.
- ✓ Una vez encendido, habrá un interruptor de selección, cuando el interruptor esté en un estado OFF el microcontrolador no activará el funcionamiento del módulo Bluetooth ni del Datalogger. Cuando el interruptor esté en modo ON, el microcontrolador estará atento a la llegada de datos por Bluetooth.
- ✓ En el LCD se muestra el estado de los procesos que se realizan en el hardware.
- ✓ El PIC está constantemente esperando los identificadores que fueron mencionados en el capítulo anterior, es decir "\$" y los otros identificadores para saber cuáles son los datos va a recibir.
- ✓ Los datos llegan por Bluetooth al PIC y son enviados por él hacia el Datalogger en donde se almacenan los resultados en un archivo .txt dentro de una carpeta con el nombre de la mesa de votación
- ✓ Finaliza el proceso de envío de datos y resultados.

5.4. DIAGRAMA DE FLUJO

En la Figura 40 se muestra el diagrama de flujo correspondiente a las actividades del hardware de microcontrolador.

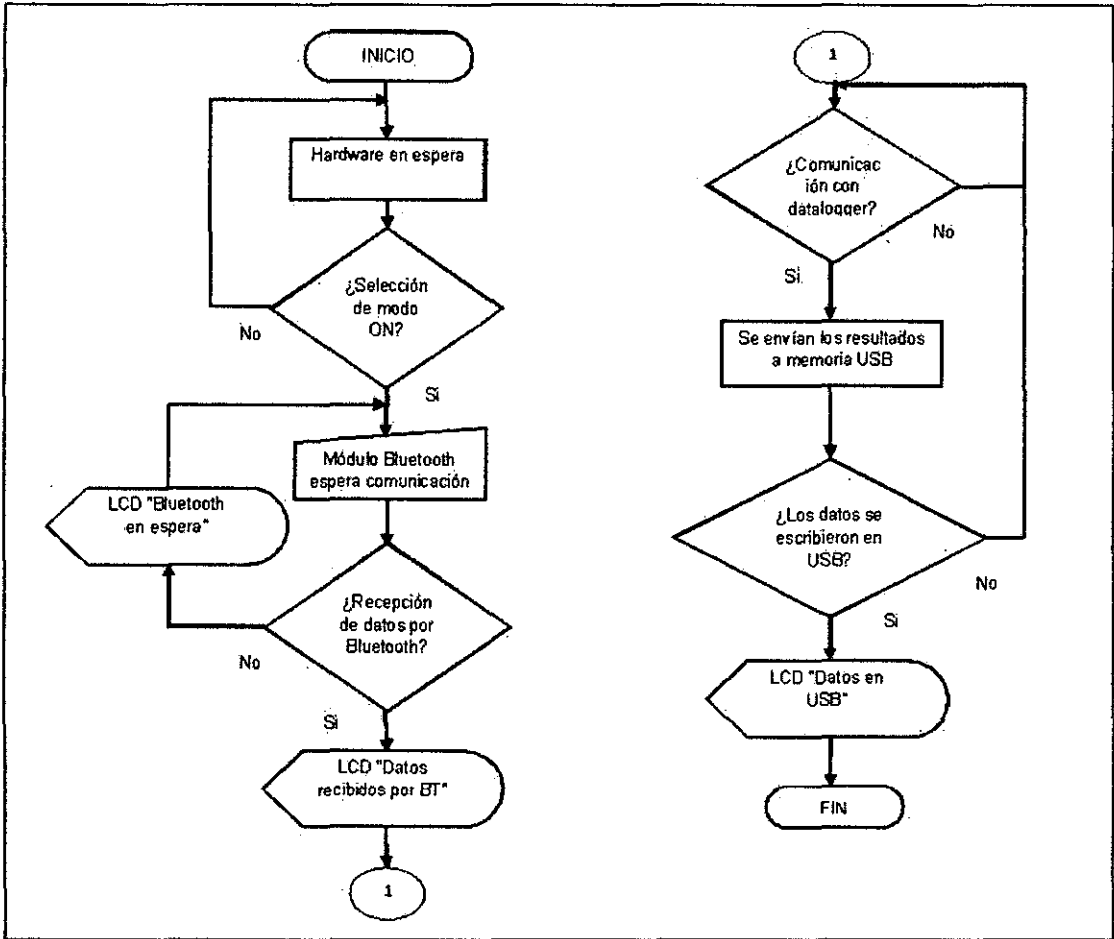


Figura 40 Diagrama de flujo del hardware

5.5. DISEÑO

Como se ha mencionado en los apartados previos, el diseño de este hardware o circuito electrónico está controlado principalmente por un microcontrolador PIC, por razones de cantidad de puertos y capacidad para manejar protocolos serie, se ha seleccionado trabajar con el PIC16F877a (Ver pines en Figura 41). Se aprovechará las ventajas mencionadas para comunicar vía serie al Datalogger y el módulo Bluetooth con el PIC, además del LCD para mostrar cada proceso realizándose en el microcontrolador con respecto a la adquisición y almacenamiento de los resultados. El LCD se ha

conectado al puerto D del PIC con uso de 4 pines de datos, el datalogger (que posee 8 pines, 4 de conexión serie) se conecta al puerto C y finalmente el módulo Bluetooth de 4 pines (2 de conexión serie) se conecta al puerto B. El interruptor para selección del modo recepción de datos se conecta en un pin del puerto D. En la Figura 42 se presenta las conexiones del PIC con los dispositivos electrónicos.

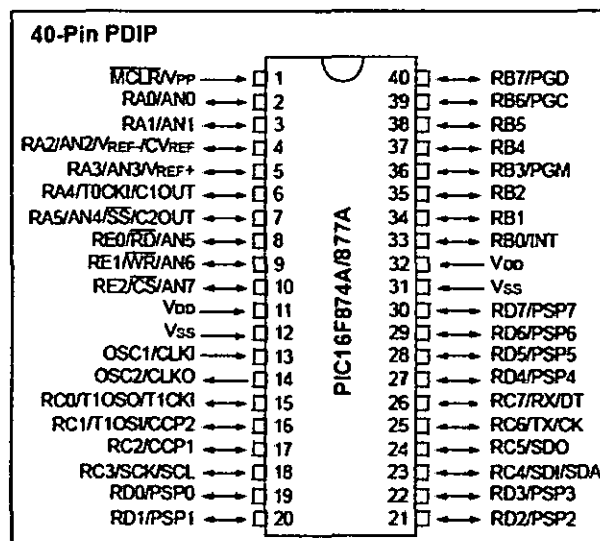


Figura 41 Configuración de pines PIC16F877a [29]

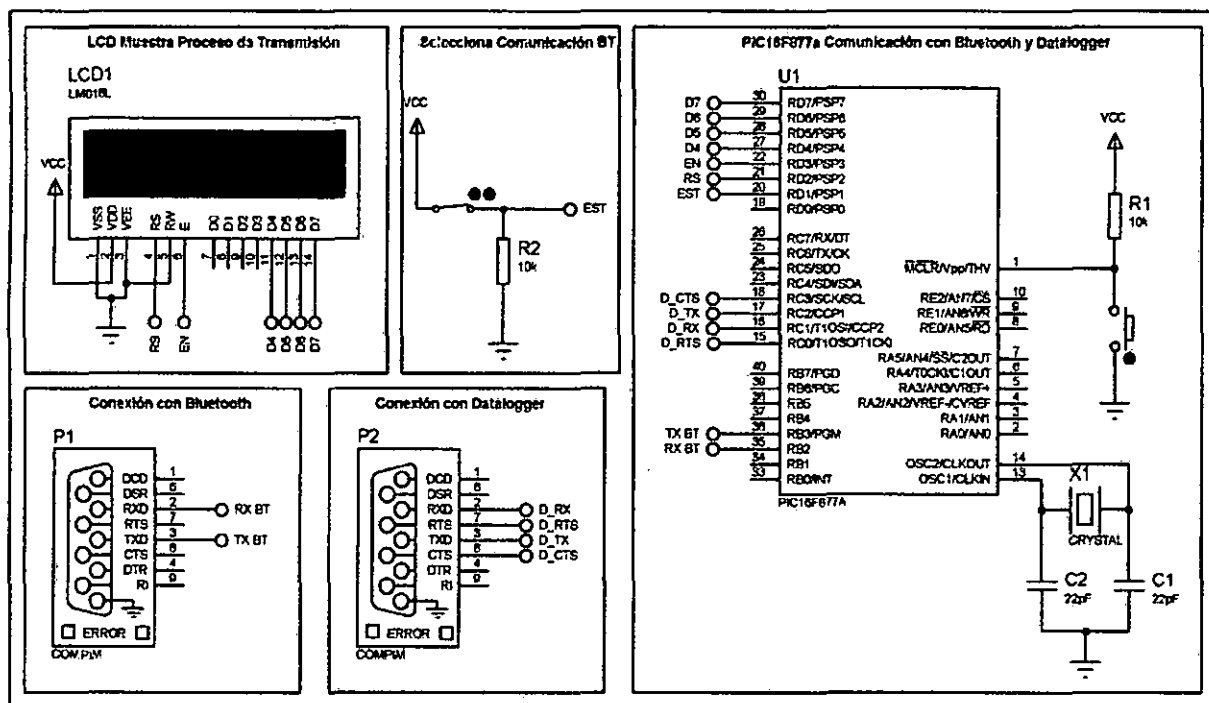


Figura 42 Conexiones y circuito del diseño (Imagen referencial)

### 5.5.1. MÓDULO BLUETOOTH

Para este proyecto se ha optado por usar el módulo *Bluetooth Slave HC-06* el cual se muestra en la Figura 43. Con dimensiones tan pequeñas [30] como un conector USB, el módulo Bluetooth HC-06 es una de las piezas de mercado más económicas que puedas encontrar dedicadas a este tipo de conectividad. La mayoría de los aficionados a los enlaces inalámbricos de datos, encuentran en este pequeño dispositivo todo lo necesario para resolver sus conexiones "wireless" y a un costo que no alcanza a la quinta parte de lo que hay que pagar por otros dispositivos de las mismas características.

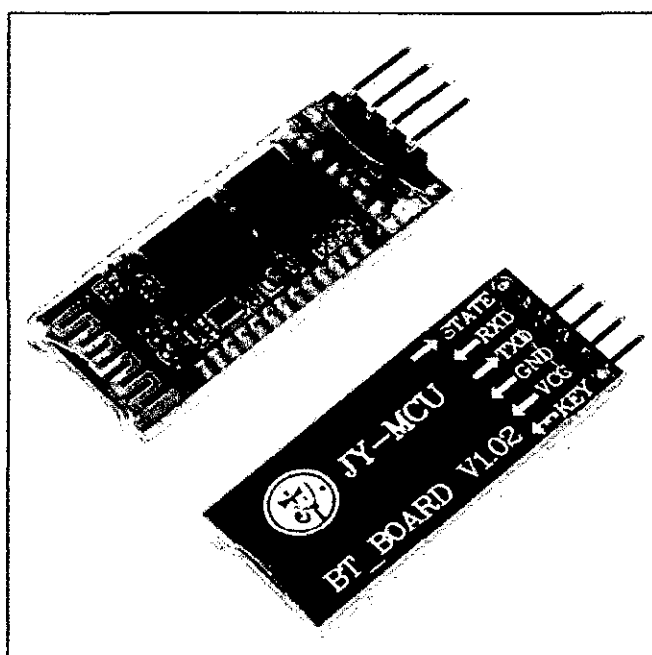


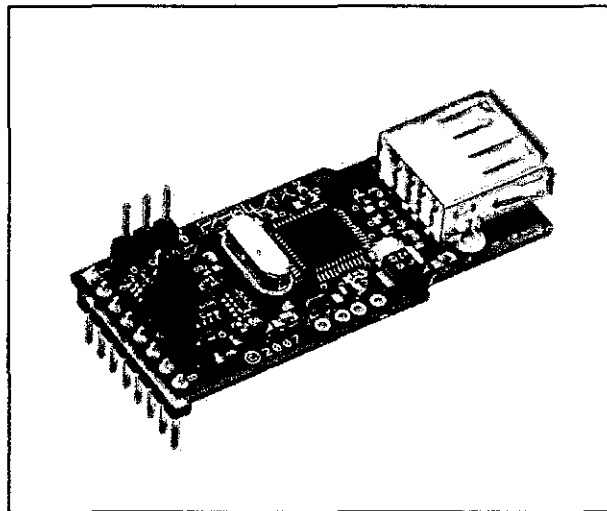
Figura 43 Módulo Bluetooth HC-06 [30]

La principal ventaja de este módulo es que su tensión de alimentación es de 3,3Volts y tiene bajo consumo (8mA en transmisión/recepción activa), lo que lo transforman en un dispositivo ideal para trabajar con microcontroladores de la misma tensión de alimentación, logrando de este modo equipos portátiles que pueden ser alimentados durante muchas horas por baterías recargables o alcalinas AA.

Como modo de fábrica, este dispositivo viene configurado para trabajar como esclavo, a 9600 baudios, sin paridad, 8 bits, 1 bit de parada y pincode por defecto "1234".

### 5.5.2. MÓDULO DATALOGGER

Como se ha mencionado párrafos arriba, para el registro y/o almacenamiento de los resultados del escrutinio en el hardware, se va a utilizar un datalogger, en este proyecto se ha seleccionado el *Parallax Memory Stick Datalogger* el cual se puede apreciar en la Figura 44. El Memory Stick Datalogger [31] es un conector USB host que permite conectar un dispositivo USB de almacenamiento masivo, como una unidad flash, a un Basic Stamp, SX o a otro microcontrolador. El firmware/chip Vinculum en el módulo registrador de datos maneja el sistema de archivos de la Memory Stick para que pueda compartir los archivos con una PC. Todo este control mediante comandos seriales simples. Este dispositivo es ideal para el registro remoto de grandes cantidades de datos, y almacenamiento de bases de datos de RFID u otras aplicaciones.



**Figura 44** Parallax Memory Stick Datalogger [31]

Este dispositivo presenta las siguientes características:

- ✓ Interfaz serial simple o SPI para comunicar con microcontrolador.
- ✓ Comandos de configuración y respuesta en formato corto y extendido.
- ✓ Alimentación a 5V.
- ✓ Fácil manipulación y constantes actualizaciones del Firmware.

### 5.5.3. CÓDIGO DEL MICROCONTROLADOR

Siguiendo referencia de los comandos que se pueden encontrar en la página web del *Parallax Memory Stick Datalogger*, y sabiendo que el módulo Bluetooth HC-06 se puede configurar con comandos AT enviados desde el PIC; se opta por dejar configurado el módulo Bluetooth por defecto con el nombre de **HC-06** que al ser del tipo slave, está configurado para esperar las órdenes que vendrán desde el programa de Android.

Como primer aspecto, los datos provenientes de la aplicación serán los 6 dígitos de la mesa de votación establecida en el proceso electoral, y los resultados del escrutinio por cada candidato (en este proyecto como prototipo, 8 postulantes) donde cada resultado tiene 3 dígitos y es recibido por el HC-06 almacenándolo en una variable tipo Word para luego enviarlo al datalogger. En la Figura 45 se puede apreciar los comandos principales para la recepción de datos vía Bluetooth. El microcontrolador ha sido programado en *PicBasic Pro* con el entorno *Microcode Studio*.

```
-----[Recepción Bluetooth]-----
MESA1:
IF (SELECCION = 1) THEN
  LCDOUT $FE, 1, "RECEPCION LISTA"
  LCDOUT $FE, $C0, "ESPERANDO DATOS"
  PAUSE 200
  SERIN2 BT_RX, Baud, 10000, MESA1, [WAIT("$"), DEC1 ESPERA, DEC1 MES1, _
  DEC1 MES2, DEC1 MES3, DEC1 MES4, DEC1 MES5, DEC1 MES6]
  PAUSE 200
  IF (ESPERA = 0) THEN
    SERIN2 BT_RX, Baud, 10000, MESA1, [WAIT("&"), DEC3 RES1, DEC3 RES2, _
    DEC3 RES3, DEC3 RES4, DEC3 RES5, DEC3 RES6, DEC3 RES7, DEC3 RES8]
    PAUSE 200
    LCDOUT $FE, 1, "RECIBIENDO DATOS"
    LCDOUT $FE, $C0, " MESA ", DEC1 MES1, DEC1 MES2, DEC1 MES3, DEC1 MES4, _
    DEC1 MES5, DEC1 MES6
    PAUSE 1000
    GOSUB GRABAR_VOTOS
    GOTO MESA1
  ENDIF
  IF (ESPERA = 1) THEN
    LCDOUT $FE, 1, "RECIBIENDO DATOS"
    LCDOUT $FE, $C0, "DNI QUE NO VOTO"
    PAUSE 1000
    GOSUB GRABAR_DNI
  ENDIF
ELSE
  LCDOUT $FE, 1, " SELECCIONE"
  LCDOUT $FE, $C0, " SWITCH"
  PAUSE 200
  GOTO MESA1
ENDIF
```

Figura 45 Configuración y recepción Bluetooth del PIC



Como se estableció en el apartado 4.4.2 la comunicación entre la aplicación Android y el microcontrolador PIC se hará a través del módulo Bluetooth HC-06 el cual esperará el símbolo “\$” con el dígito ESPERA y con los dígitos de la mesa de votación. Dependiendo del valor de la variable ESPERA que puede ser 1 o 0 se procederá a llamar a la correspondiente subrutina. Si ESPERA es 0, el PIC espera recibir el identificador “&” seguido de los resultados de conteo de votos por candidato en formato de tres dígitos, muestra el proceso en el LCD y manda a la subrutina **GRABAR\_VOTOS**. Si ESPERA es 1, el PIC indica en el LCD que estará escribiendo los DNIs que no realizaron votación y por ende, manda a la subrutina **GRABAR\_DNI**.

Se destaca que la conexión entre el módulo Bluetooth y el PIC es del tipo UART o serial asíncrono, y se trabaja en picbasic pro con el comando **SERIN2** y **SEROUT2** a 9600 baudios.

Cuando se llame a las subrutinas, se enviarán los resultados al datalogger para almacenarlos en dos archivos de texto llamados **VOTOS.TXT** y **NOVOTOS.TXT**, que se almacenarán en una carpeta con el nombre del número de la mesa de votación de la cual se hayan enviado los datos. Para evitar la complejidad de la transmisión de datos, se trabaja solamente con los números de mesa y resultados de votación, así de esta manera se espera que el archivo de texto luego pueda ser leído e interpretado en un computador de mayor capacidad.

Como segundo aspecto, dentro del mismo programa del microcontrolador se ha diseñado la parte correspondiente al datalogger. Se debe mencionar que los comandos principales para controlar el datalogger de **Parallax** están especificados en la misma página del fabricante descargables de [31]. Los principales comandos para la configuración del datalogger son:

- De sincronización. Se envía la letra “E” y “e” como comando de eco para comprobar la conectividad entre PIC y datalogger.
- Comando MKD. Creación de carpeta en la memoria USB.
- Comando CD. Apertura de la carpeta y regreso al directorio raíz.

- Comando OPW. Crea un archivo con el nombre que se indique después de él y lo abre para escritura.
- Comando WRF. Escribe dígitos y/o número en el archivo de texto creado por el comando OPW.
- Comando CLF. Cierra el archivo. Finalización de escritura.

```

-----[Sincronizando Datalogger]-----
LCDOUT $FE, 1, " Inicializando"
LCDOUT $FE, $C0, " Datalogger-USB"
PAUSE 2000
'Iniciar linea
HIGH TX
LOW RTS
PAUSE 2000
'Sincronizar
GOSUB Purge
LCDOUT $FE, 1, " Sincronizando "
LCDOUT $FE, $C0, " Datalogger-USB"
index = 0
'Para sincronizacion enviar E hasta recibir eco
WHILE (index < 1)
  PAUSE 500
  SEROUT2 TX\CTS, Baud, {"E"}
  SEROUT2 TX\CTS, Baud, {13}
  GOSUB Get_Serial_Bytes
WEND
'Enviar e para finalizar sincronización
PAUSE 500
SEROUT2 TX\CTS, Baud, {"e"}
SEROUT2 TX\CTS, Baud, {13}
GOSUB Get_Serial_Bytes
'Enviar CR si se encuentra el dispositivo
PAUSE 500
GOSUB Purge
LCDOUT $FE, 1, "Esperando al USB"
LCDOUT $FE, $C0, " ....."
SEROUT2 TX\CTS, Baud, {13}
GOSUB Get_Serial_Bytes
'Detectando USB
'Purgar Buffer
'Enviar Carriage Return
'Esperar por D:\>

```

**Figura 46** Sincronización de Datalogger

En la Figura 46 se aprecia parte del código usado para sincronización del datalogger. Se han enviado las letras “E” y “e” para detectar si se encuentra conectado el dispositivo y el dígito “13” para reconocer si se tiene o no conectada la memoria USB. Este proceso se lleva a cabo al iniciar el programa del PIC una vez alimentado a 5V.

En la Figura 47 se muestra parte del código para la escritura de los datos en la memoria USB, donde después de recibir los datos por Bluetooth se llama a la subrutina GRABAR\_VOTOS para enviar los resultados del escrutinio al datalogger.

```

-----[Grabando Datalogger]-----
GRABAR_VOITOS:
  'Comandos cortos datalogger
  PAUSE 200
  SEROUT2 TX\CTS, Baud, ["ECS", 13]
  GOSUB Get_Serial_Bytes
  'Regresa al directorio Raiz
  PAUSE 200
  SEROUT2 TX\CTS, Baud, ["CD ..", 13]
  GOSUB Get_Serial_Bytes
  'Crear folder
  PAUSE 200
  SEROUT2 TX\CTS, Baud, ["MKD ", DEC1 MES1, DEC1 MES2, DEC1 MES3, DEC1 MES4, _
  DEC1 MES5, DEC1 MES6, 13]
  GOSUB Get_Serial_Bytes
  'Entrar al folder
  PAUSE 200
  SEROUT2 TX\CTS, Baud, ["CD ", DEC1 MES1, DEC1 MES2, DEC1 MES3, DEC1 MES4, _
  DEC1 MES5, DEC1 MES6, 13]
  GOSUB Get_Serial_Bytes
  'Si el archivo existe lo borra para actualizar
  PAUSE 200
  SEROUT2 TX\CTS, Baud, ["DLF VOTOS.TXT", 13]
  GOSUB Get_Serial_Bytes
  'Crear archivo .txt
  PAUSE 200
  SEROUT2 TX\CTS, Baud, ["OPW VOTOS.TXT", 13]
  GOSUB Get_Serial_Bytes
  'Escribir en el archivo
  PAUSE 200
  SEROUT2 TX\CTS, Baud, ["WRF ", $00, $00, $00, $26, 13, DEC3 RES1, 13, 10, _
  DEC3 RES2, 13, 10, DEC3 RES3, 13, 10, DEC3 RES4, 13, 10, DEC3 RES5, 13, 10, _
  DEC3 RES6, 13, 10, DEC3 RES7, 13, 10, DEC3 RES8, 13]
  GOSUB Get_Serial_Bytes
  'Cierra el archivo
  PAUSE 200
  SEROUT2 TX\CTS, Baud, ["CLF VOTOS.TXT", 13]
  GOSUB Get_Serial_Bytes
  'Regresa al directorio Raiz
  PAUSE 200
  SEROUT2 TX\CTS, Baud, ["CD ..", 13]
  GOSUB Get_Serial_Bytes
  'Termina Subrutina
  RETURN

```

**Figura 47** Subrutina para enviar el conteo de votos al Datalogger

Claramente, se ve en la Figura 47 el uso de los comandos mencionados antes y se resalta el comando WRF que escribe los datos en sí de los resultados del escrutinio en un archivo de texto llamado VOTOS.TXT dentro de una carpeta que lleva por nombre los dígitos de la mesa de votación correspondiente.

Para el caso en que el PIC reciba los datos de los DNIs que no realizaron votación, se llama a la subrutina **GRABAR\_DNI**. Aquí, el PIC estará atento a dos situaciones, una es que reciba el identificador “%” seguido de los 4 primeros dígitos y los 4 segundos dígitos del primer DNI que no votó, la otra es que reciba la variable

CONFIRMA que será 1 si es que la aplicación tiene que seguir enviando DNIs o será 0 si es que ya no habrá que esperar a que lleguen más datos. Esto es así porque pueden haber pocas o muchas personas que no hayan realizado votación y, para poder grabar todas a través del datalogger, se ha preferido enviar uno a uno los DNIs correspondientes a cada persona. De esta forma la subrutina se mantendrá mientras haya datos por recibir y actualizará automáticamente el archivo de texto NOVOTOS. En la Figura 48 se puede ver parte del código para escribir los datos de los DNIs.

```
GRABAR_DNI:
'Comandos cortos datalogger
PAUSE 200
SEROUT2 TX\CTS, Baud, ["ECS", 13]
GOSUB Get_Serial_Bytes
'Regresa al directorio Raiz
PAUSE 200
SEROUT2 TX\CTS, Baud, ["CD ..", 13]
GOSUB Get_Serial_Bytes
'Crear folder
PAUSE 200
SEROUT2 TX\CTS, Baud, ["MKD ", DEC1 MES1, DEC1 MES2, DEC1 MES3, DEC1 MES4, _
DEC1 MES5, DEC1 MES6, 13]
GOSUB Get_Serial_Bytes
'Entrar al folder
PAUSE 200
SEROUT2 TX\CTS, Baud, ["CD ", DEC1 MES1, DEC1 MES2, DEC1 MES3, DEC1 MES4, _
DEC1 MES5, DEC1 MES6, 13]
GOSUB Get_Serial_Bytes
'Crear archivo .txt
PAUSE 200
SEROUT2 TX\CTS, Baud, ["OPW NOVOTOS.TXT" , 13]
GOSUB Get_Serial_Bytes
'Esperamos recibir datos
PAUSE 200
SERIN2 BT_RX, Baud, 10000, GRABAR_DNI, [WAIT("%"), DEC4 DNI1, DEC4 DNI2, _
DEC1 CONFIRMA]
PAUSE 200
SEROUT2 TX\CTS, Baud, ["WRF ", $00, $00, $00, $0A, 13, DEC4 DNI1, DEC4 DNI2, _
13, 10, 13]
GOSUB Get_Serial_Bytes
'Cierra el archivo
PAUSE 200
SEROUT2 TX\CTS, Baud, ["CLF NOVOTOS.TXT", 13]
GOSUB Get_Serial_Bytes
'Regresa al directorio Raiz
PAUSE 200
SEROUT2 TX\CTS, Baud, ["CD ..", 13]
GOSUB Get_Serial_Bytes
PAUSE 200
'Condiciones
IF (CONFIRMA = 1) THEN
    GOSUB GRABAR_DNI
ENDIF
IF (CONFIRMA = 0) THEN
    GOTO MESA1
ENDIF
```

Figura 48 Subrutina para enviar DNIs al datalogger

De esta manera se graban los datos del conteo de votos total por mesa de los candidatos así como los DNIs de cada persona que no haya participado en la elección. Pues se ha considerado que estos datos son los de mayor requerimiento y prisa en un proceso electoral. Se agiliza el proceso de conteo ya que es realizado por la misma aplicación y se tiene también un archivo de texto que incluye a las personas que no votaron para la posterior sanción por parte del organismo regulador. Estos archivos podrían ser luego leídos por alguna aplicación de computador.

## **5.6. RESUMEN**

Un hardware o circuito electrónico controlado principalmente por un microcontrolador PIC para la adquisición y almacenamiento de datos (resultados) del escrutinio o proceso electoral ha sido diseñado. Dicho hardware permite comodidad para tener los datos a mano en el menor tiempo posible y facilidad para transportarse como dispositivo de bolsillo. En este capítulo:

- ✓ Se ha diseñado un hardware basado en el microcontrolador PIC16F877a que a través de su puerto B se conecta a un módulo Bluetooth HC-06 para recibir los datos y resultados del proceso electoral provenientes de la aplicación de votación para Android realizada en el capítulo IV. Los datos son recepcionados y almacenados en la memoria interna del PIC. En un LCD conectado al puerto D se muestran los estados de los procesos a realizarse. Por ejemplo si el hardware está recibiendo datos vía Bluetooth.
- ✓ También el microcontrolador PIC se conecta a un módulo Parallax Memory Stick Datalogger a través de su puerto C para almacenar los datos recibidos por Bluetooth en una memoria USB en la cual se guardan los resultados en un archivo de texto .txt que contiene los candidatos y su cantidad de votos. En esta memoria USB se guardará tanto los resultados de votación como las personas que no hayan realizado el escrutinio en el tiempo establecido por el organismo regulador. Nuevamente se usa el LCD como medio controlador de los procesos del hardware.

## CAPÍTULO VI

### PROTOTIPO Y SU FUNCIONAMIENTO

Después de haber presentado el diseño tanto del software como del hardware en los capítulos IV y V, se presentará en éste la implementación de un prototipo real. En este contexto, se ha exportado la aplicación Android desde Eclipse en un archivo .apk para instalarlo en una tableta (el archivo pesa 258 KB), se ha construido el hardware comandado por el microcontrolador PIC con las conexiones mostradas anteriormente al módulo Bluetooth y al módulo Datalogger. También se decidió usar como tableta de prueba la Samsung Tab 2 (7.0) que trae Android versión 4.0, pantalla de 7 pulgadas y 1GB de RAM.

#### 6.1. VOTACIÓN

Una vez instalada la aplicación en la tableta, se procedió a las pruebas. En la Figura 49 se muestra la actividad principal a la cual se le llamó *configuración de mesa*. Se ha ingresado el número de mesa y la contraseña (datos especificados en las bases de datos, datos no reales y con propósito de pruebas).

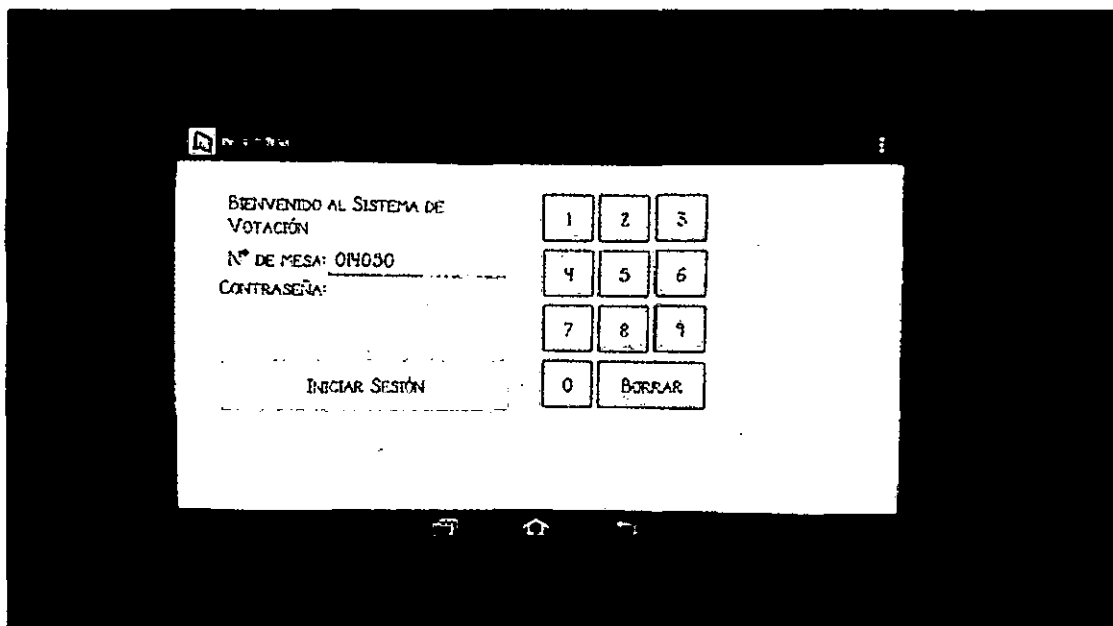


Figura 49 Inicio de la aplicación

Una vez el programa hizo la corroboración en la base de datos de la mesa y la contraseña, muestra la segunda actividad llamada *ingreso de DNI*. La aplicación realiza dos comprobaciones, si el DNI que se ingresa pertenece a la mesa de votación en funcionamiento y si dicho DNI no ha realizado votación anteriormente. Luego de las comprobaciones correspondientes se muestra la tercera actividad llamada *selección de candidato*. En las Figura 50 y 51 se aprecian las actividades.

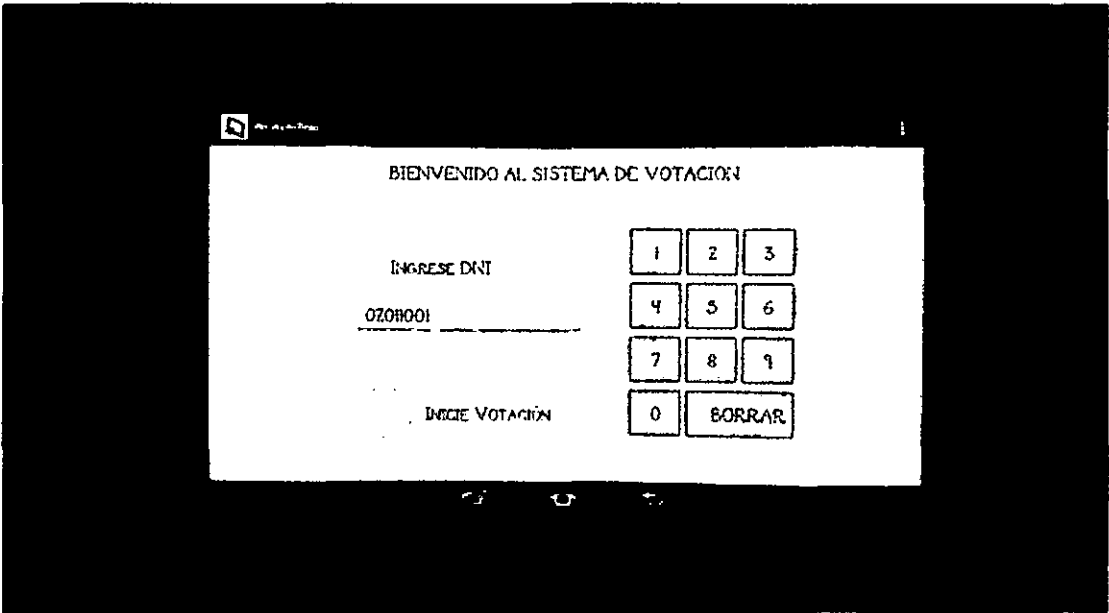


Figura 50 Pantalla de ingreso de DNI

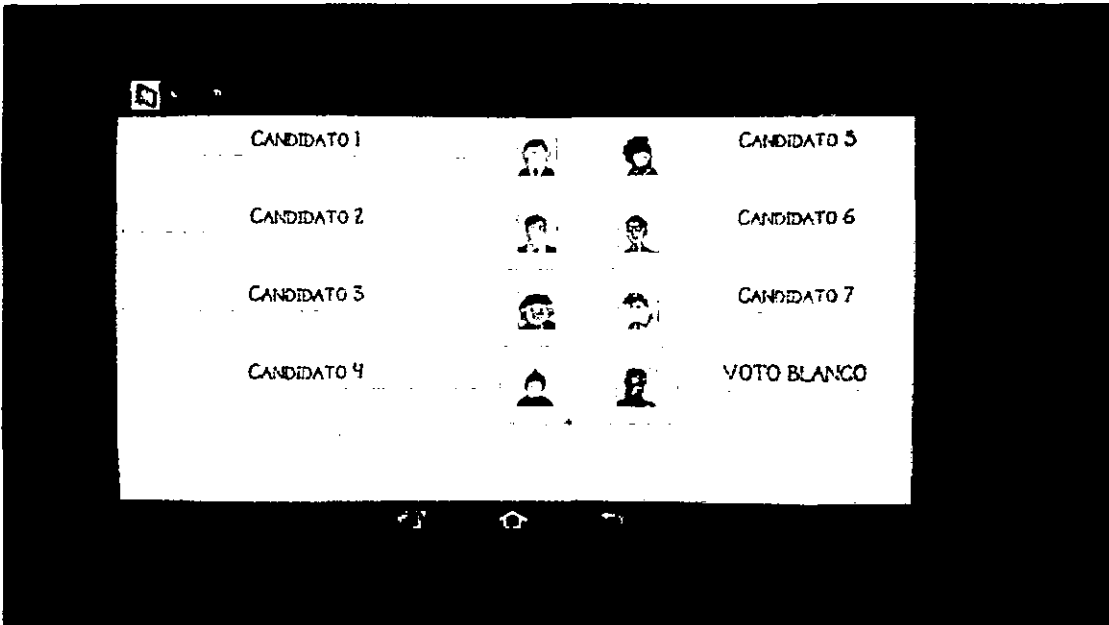


Figura 51 Cédula de votación

En la cédula de votación el votante puede presionar el nombre de su candidato de preferencia y se confirmará su votación. Además, como se detalló en los capítulos de diseño la aplicación debe ser capaz de reconocer si el votante ya había realizado su voto. En la Figura 52 se muestra la pantalla de confirmación de votación y en la Figura 53 la pantalla que se muestra cuando una persona que ya realizó su voto intenta hacerlo nuevamente.

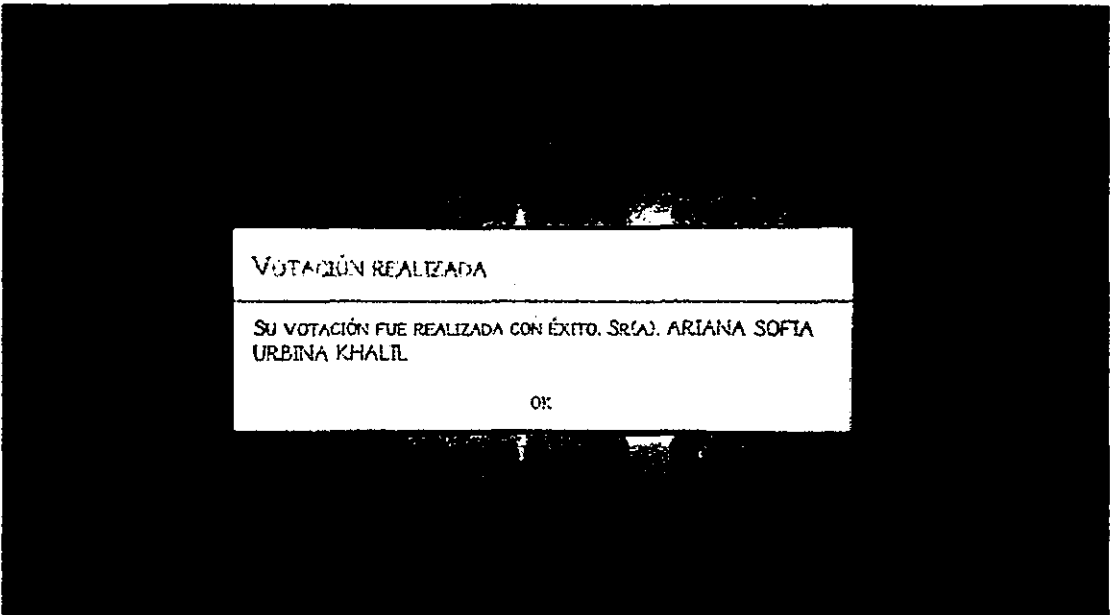


Figura 52 Confirmación de votación

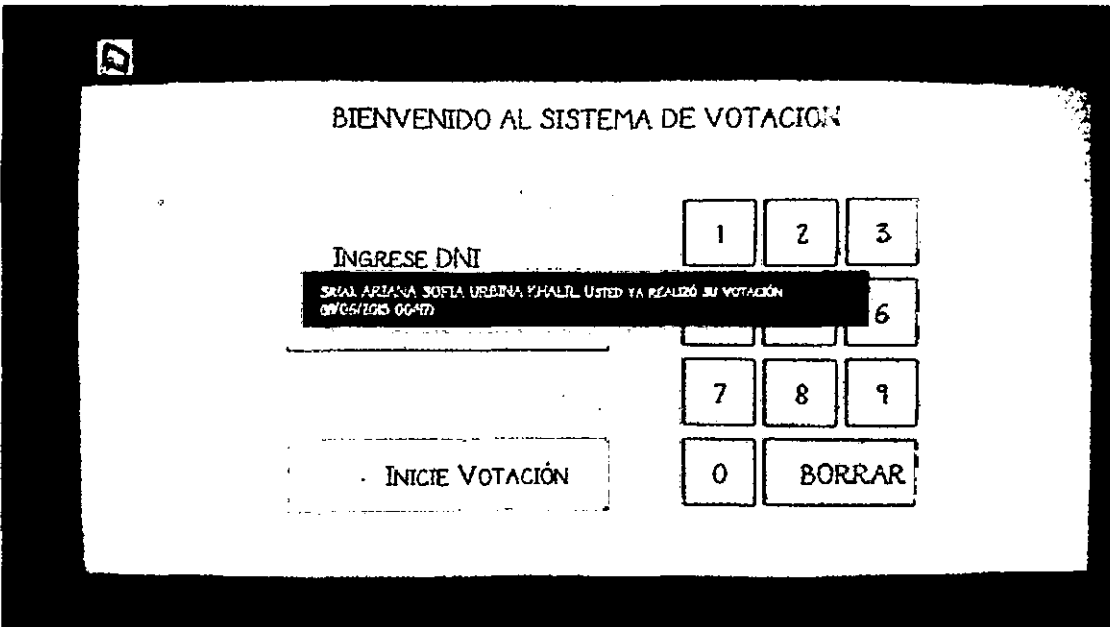


Figura 53 Indicación de que ya se había realizado el voto



## 6.2. ENVÍO DE LOS DATOS

Una vez finalizado el proceso electoral, se accede al menú de opciones para enviar los datos mediante el Bluetooth al hardware que a través de un microcontrolador PIC 16F877A recibe los datos con un módulo HC-06 y los datos recibidos son grabados en un archivo de texto en una memoria USB a través de un datalogger de Parallax.

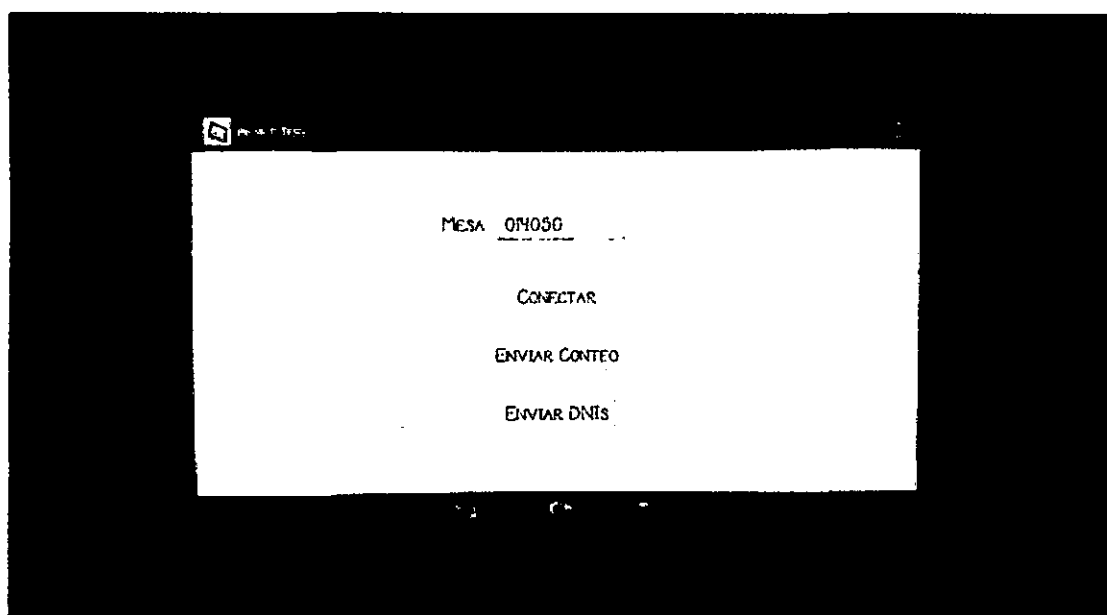


Figura 54 Actividad de envío de datos

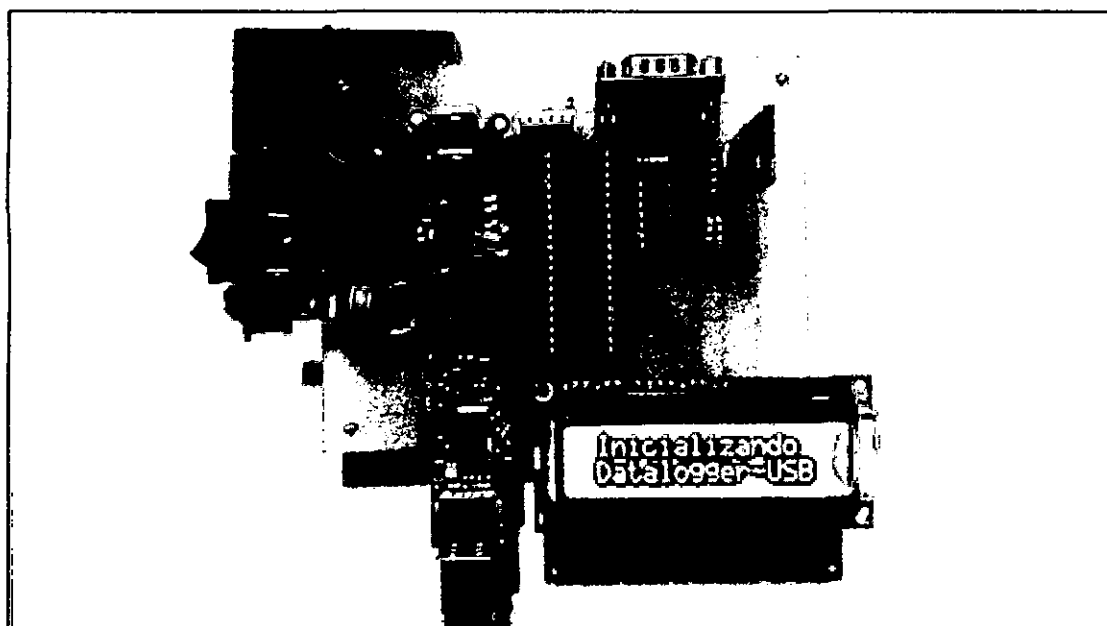


Figura 55 Inicializando y sincronizando el datalogger

Se puede apreciar en la Figura 54 que para este ejemplo se enviaron los datos de la supuesta mesa 014050 y se muestra la pantalla de conexión Bluetooth. En la Figura 55 se observa el inicio del hardware en donde el PIC inicializa y sincroniza al módulo datalogger. Una vez listo el hardware (Ver Figura 56), podemos conectar la tableta con él presionando el botón *Conectar* (Ver Figura 54).

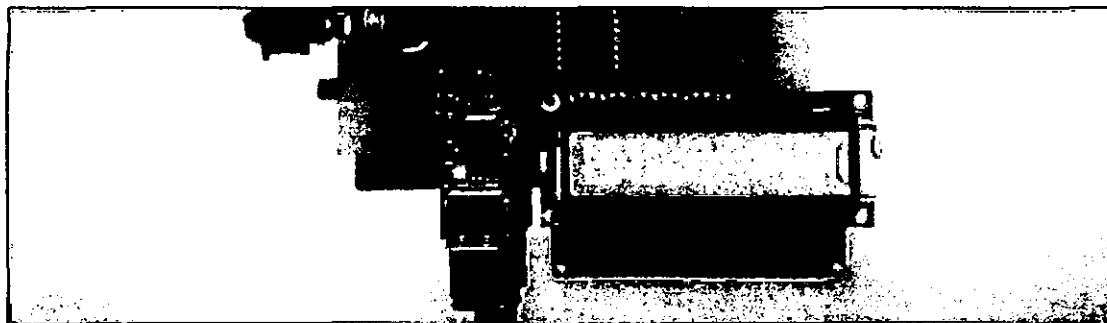


Figura 56 Hardware listo

Al presionar el botón de conectar la tableta se conecta al HC-06 el cual tiene un led rojo que deja de parpadear (parpadea cuando no está vinculado). Presionamos el botón de *Enviar Conteo* y esperamos hasta que el LCD indique otra vez que la recepción está lista. Presionamos el botón *Enviar DNIs* y otra vez esperamos que el LCD nos indique que acabó la transmisión de datos. En las Figura 57 y 58 se muestran estos procesos indicados por el Hardware.

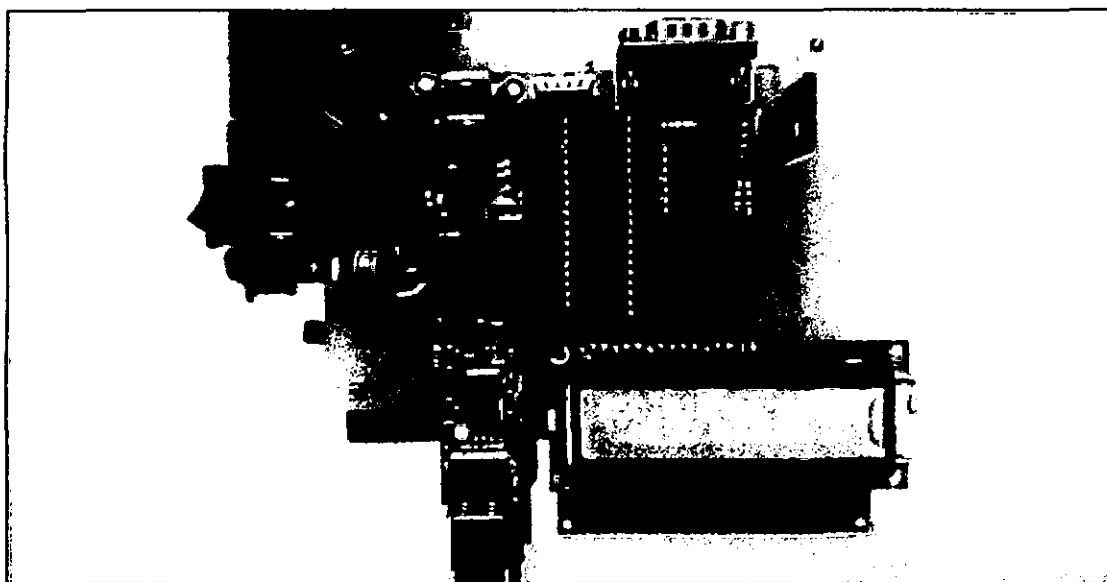


Figura 57 Se reciben los datos de los conteos por candidato

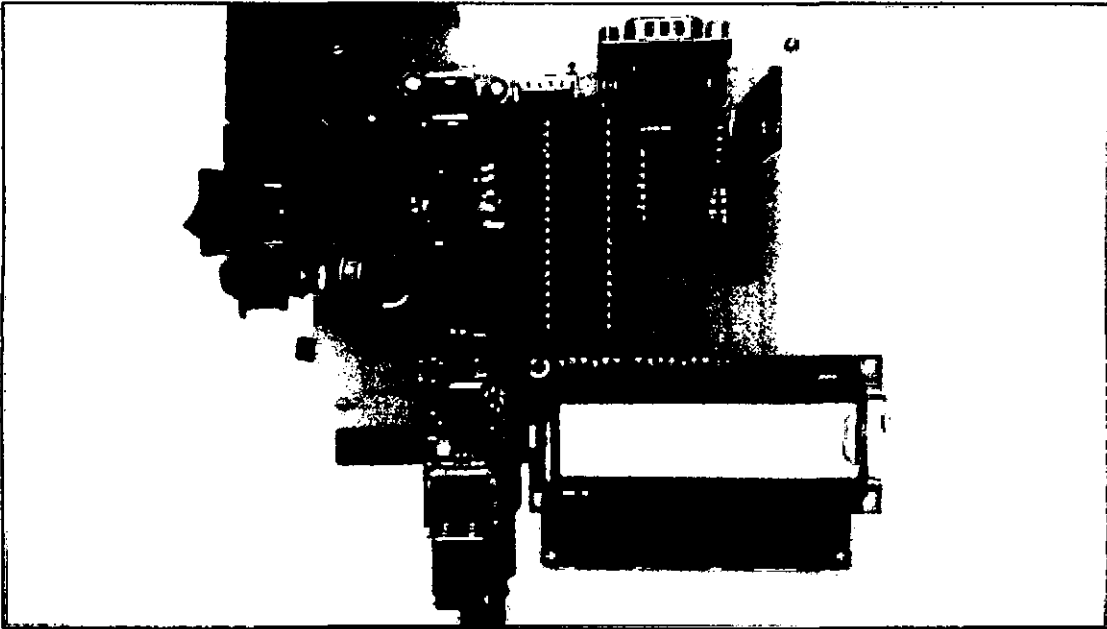


Figura 58 Se reciben los datos de los DNIs que no realizaron votación

Finalmente se ha revisado la memoria USB en un computador para ver si los datos se grabaron correctamente. Los resultados demuestran que se logró transmitir tanto los datos resultantes del escrutinio como los DNIs de las personas que no votaron. En las Figuras 59 y 60 se muestran los datos grabados.

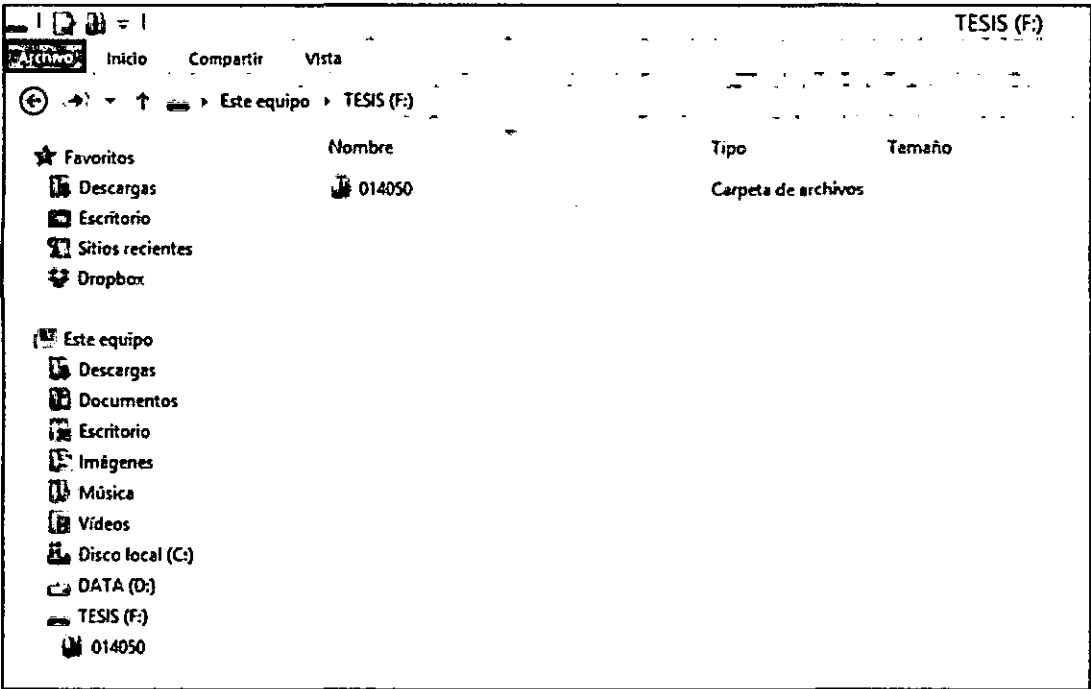


Figura 59 Carpeta de la mesa de votación

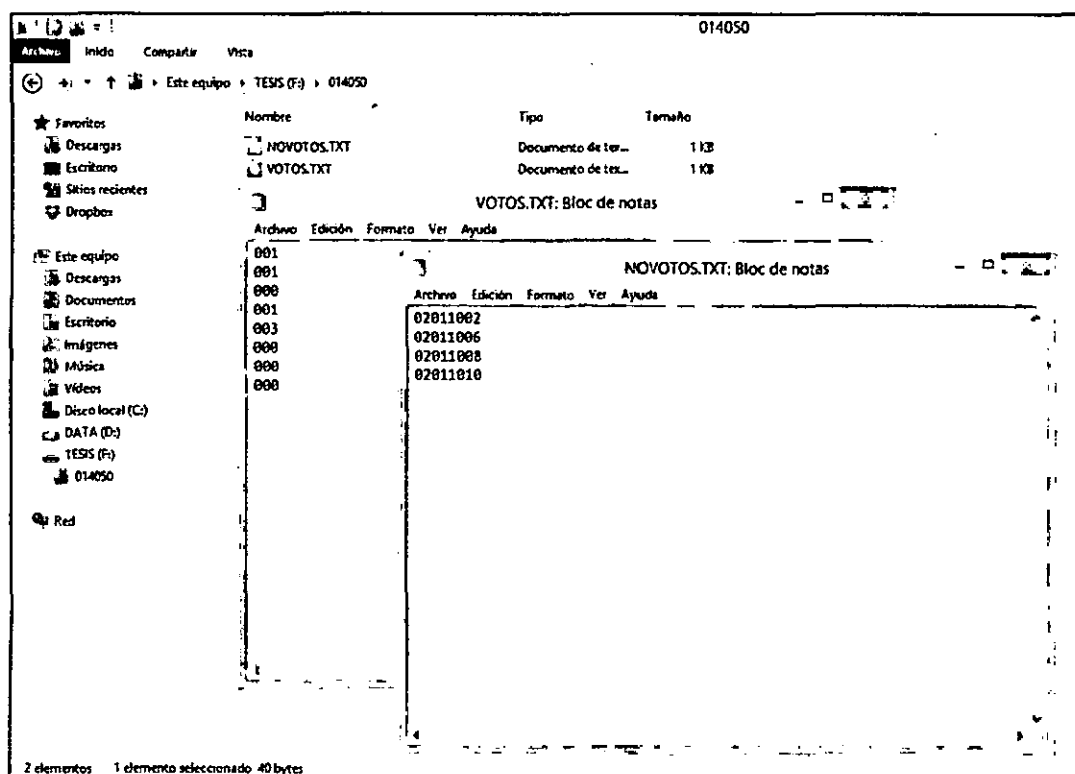


Figura 60 Archivos de texto que contienen los datos enviados

### 6.3. COSTOS

Los precios de los componentes y dispositivos usados en la implementación son:

COMPONENTE	COSTO (Soles)
Datalogger Parallax	110.00
HC-06 Bluetooth	25.00
PIC 16F877A	20.00
LCD 16x2	12.00
Resistencias, capacitores, reguladores, cristal, USB y otros	25.00
Tableta Samsung Tab 2	480.00
<b>TOTAL</b>	<b>672.00</b>

## **CAPÍTULO VII**

### **CONCLUSIONES**

En esta tesis se ha diseñado un prototipo de un sistema de votación electrónica creado para el sistema operativo Android que junto con un circuito electrónico controlado por un microcontrolador PIC, reemplazan el convencional sistema de votación en papel así como los sistemas actuales de votación electrónica contruidos con microprocesadores y computadoras.

Habiendo presentado en los capítulos IV y V el diseño de la aplicación en Eclipse del sistema de votación por tableta como cédula, la comunicación Bluetooth entre la aplicación Android y el hardware-microcontrolador, y el diseño del hardware basado en un PIC con los módulos periféricos de adquisición de datos (Bluetooth) y almacenamiento de ellos (Datalogger); y habiendo presentado el funcionamiento del prototipo en el capítulo VI; se puede concluir que:

- ✓ Se ha logrado diseñar la aplicación de votación electrónica que era el objetivo principal de esta tesis. Y se ha diseñado bajo la plataforma Java para ser utilizada en un dispositivo electrónico como una tableta, con lo que se logra poder reemplazar los sistemas convencionales de papel y las máquinas electrónicas basadas en computadoras.
- ✓ Se ha logrado diseñar e implementar un hardware comandado por un microcontrolador PIC que se comunica por vía Bluetooth con la aplicación (módulo Bluetooth del dispositivo tableta) usando un periférico Bluetooth HC-06 que recibe los datos desde la aplicación y los almacena en el Parallax Memory Stick Datalogger. La información se guarda en una memoria USB en formato .txt para poder ser transportada fácilmente.
- ✓ Con la aplicación de votación electrónica en Android y la tableta, se podrá realizar todo el proceso electoral completo. La configuración de la mesa de votación, el ingreso del DNI del votante, la selección del candidato. Como principal ventaja de este proyecto se tiene el evitar el uso de impresiones, hojas

de papel, bolígrafos, tinta, etc. Cada paso queda registrado en una base de datos que solo puede ser manipulada por medios informáticos externos como computadoras.

- ✓ Como consecuencia directa del software y hardware diseñados, se proporciona mayor seguridad al ciudadano y mayor confianza que sus votos no podrán ser modificados o anulados, los datos solo podrán ser revisados en las oficinas principales del organismo regulador. También se logra acelerar el proceso del conteo de votos, ya que no es necesario tener a los miembros de mesa contándolos sino que la aplicación calcula la cantidad de votos por candidato al finalizar el escrutinio.
- ✓ Finalmente, al ser una aplicación para Android implementada para su uso en tableta y al tener un hardware controlado por microcontrolador, en vez de tener máquinas basadas en computadoras, se logra reducir notablemente las dimensiones de los dispositivos e instrumentos electrónicos para realizar las elecciones. Esto además provee mayor facilidad para el transporte de los mismos haciendo que el sistema de votación pueda llegar a cualquier lugar del país.

## **CAPÍTULO VIII**

### **RECOMENDACIONES**

- ✓ Con respecto a la parte de software, diseño de la aplicación, se debe resaltar que se podría mejorar la parte de transmisión de los datos, en el sentido de la seguridad dado que el sistema es susceptible y vulnerable a ataques por medio de la interceptación en la parte de la comunicación de datos, desarrollando métodos de encriptación para que la información se transmita con la mayor fiabilidad posible.
- ✓ Con respecto al hardware circuito electrónico diseñado con el microcontrolador PIC, debería ser construido y testeado en diversas zonas geográficas del país para determinar su robustez y su capacidad para funcionar bajo ciertas condiciones ambientales extremas, como frío o calor en exceso.
- ✓ También se puede considerar la creación de una interfaz en computadora para que recepcione los datos provenientes del hardware una vez que se tienen los datos almacenados por el Datalogger. De esta manera se evita la manipulación del archivo de texto.
- ✓ Sobre el diseño final, se podría obviar el uso de un dispositivo hardware de microcontrolador en las zonas donde haya cobertura celular 3G/4G, donde se pueden usar tabletas con capacidad para trabajar con señales de telefonía, y aprovechar esta ventaja para que la aplicación se diseñe bajo el concepto de redes, creando un canal de comunicación y envío de los resultados del escrutinio a través de Internet, conectando con la red de telefonía nacional.

## REFERENCIAS

- [1] Tuesta Soldevilla, Fernando. *El voto electrónico*. Publicado en Lima, 2004.
- [2] Urdy Chávez, Marco A.A. *DISEÑO E IMPLEMENTACIÓN DE UN EQUIPO DE VOTO ELECTRÓNICO*. Tesis de Grado. Presentado en Lima, 2012.
- [3] Oficina Nacional de Procesos Electorales (ONPE). *Servicio de elaboración de prototipo para equipo de voto electrónico*. Publicado como Exoneración N° 0003-2009-ONPE, 2009.
- [4] Oficina Nacional de Procesos Electorales (ONPE). *Historia del voto electrónico, Perú 1996-2004*. Primera Edición. Publicado en Lima, 2011.
- [5] Ministerio Británico para el Desarrollo Internacional (DFID). *Para promover la participación electoral en zonas rurales*. Publicado en Lima, 2003.
- [6] Wikipedia. *Elecciones generales de Perú de 2000*. Página web. Disponible en: [http://es.wikipedia.org/wiki/Elecciones generales de Per%C3%BA de 2000](http://es.wikipedia.org/wiki/Elecciones_generales_de_Per%C3%BA_de_2000)
- [7] Elecciones en Perú. *El JNE sustentó presupuesto para elecciones 2015*. Página web. Disponible en: <http://www.eleccionesenperu.com/noticias-presupuesto-electoral-2015-jne-2490.html>
- [8] Wikipedia. *Elecciones*. Página web. Disponible en: <http://es.wikipedia.org/wiki/Elecciones>
- [9] Oficina Nacional de Procesos Electorales (ONPE). *Voto electrónico*. Página web. Disponible en: <http://www.web.onpe.gob.pe/voto-electronico.html>
- [10] Oficina Nacional de Procesos Electorales (ONPE). *Historia del voto electrónico, Perú 2005-2012*. Primera Edición. Publicado en Lima, 2012.
- [11] Buenas Tareas. *El voto electrónico en la historia*. Página web. Disponible en: <http://www.buenastareas.com/ensayos/El-Voto-Electronico-En-La-Historia/2618826.html>
- [12] Gira BSAS. *Habrà voto electrónico en la ciudad, ¿cómo funcionará?* Página web. Disponible en: <http://www.girabsas.com/nota/2925/>
- [13] Buenas Tareas. *El voto electrónico en la historia*. Página web. Disponible en: <http://www.buenastareas.com/ensayos/El-Voto-Electronico-En-La-Historia/2618826.html>
- [14] Panizo Alonso, Luis. *Aspectos tecnológicos del voto electrónico*. Oficina Nacional de Procesos Electorales (ONPE). Primera Edición. Publicado en Lima, 2007.



- [15] Smartmatic. Página web. Disponible en: [www.smartmatic.com](http://www.smartmatic.com)
- [16] Oficina Nacional de Procesos Electorales (ONPE). *Observatorio del voto-e en Latinoamérica*. Publicado en Lima, 2011.
- [17] Smartmatic. *Elecciones Bélgica 2012 – 2014*. Página web. Disponible en: <http://www.smartmatic.com/es/experiencia/articulo/elecciones-belgica-2012-2014/>
- [18] Smartmatic. *Getting to know the next generation of voting technology: the Smartmatic Electronic Voting Machines*. Página web. Disponible en: <http://www.smartmatic.info/Sistema/saes-3370/>
- [19] Wikipedia. *Android*. Página web. Disponible en: <http://es.wikipedia.org/wiki/Android>
- [20] Wikipedia. *Tableta (computadora)*. Página web. Disponible en: [http://es.wikipedia.org/wiki/Tableta\\_\(computadora\)](http://es.wikipedia.org/wiki/Tableta_(computadora))
- [21] Java. *¿Qué es la tecnología Java y para qué la necesito?* Página web. Disponible en: [http://www.java.com/es/download/faq/whatis\\_java.xml](http://www.java.com/es/download/faq/whatis_java.xml)
- [22] Eclipse. *What is Eclipse?* Página web. Disponible en: [http://help.eclipse.org/indigo/index.jsp?topic=/org.eclipse.platform.doc.isv/guide/int\\_eclipse.htm](http://help.eclipse.org/indigo/index.jsp?topic=/org.eclipse.platform.doc.isv/guide/int_eclipse.htm)
- [23] Java Ya. *Pasos para crear un programa en Eclipse*. Página web. Disponible en: <http://www.javaya.com.ar/detalleconcepto.php?codigo=76&inicio=>
- [24] SQLite. *Welcome to SQLite*. Página web. Disponible en: <https://www.sqlite.org/>
- [25] Reyes, Carlos A. *Microcontroladores PIC 16F62X 16F81X 16F87X Programación en Basic*. Tercera Edición. Volumen 1. Publicado en Quito, 2008.
- [26] Monografías. *Control de un invernadero*. Página web. Disponible en: <http://www.monografias.com/trabajos81/control-invernadero/control-invernadero2.shtml>
- [27] Gálvez Navarro, Jaime. *Control de un módulo bluetooth mediante microcontrolador*. Trabajo fin de carrera. Presentado en Barcelona, 2005.
- [28] Wikipedia. *Registrador de datos*. Página web. Disponible en: [http://es.wikipedia.org/wiki/Registrador\\_de\\_datos](http://es.wikipedia.org/wiki/Registrador_de_datos)
- [29] Microchip. *PIC16F87xA Data Sheet 28/40/44-Pin Enhanced Flash Microcontrollers*. Publicado en Estados Unidos, 2003.

- [30] Neoteo. *Módulo Bluetooth HC-06 (Android)*. Página web. Disponible en:  
<http://www.neoteo.com/modulo-bluetooth-hc-06-android/>
- [31] Parallax. *Memory Stick Datalogger*. Página web. Disponible en: Parallax:  
<https://www.parallax.com/product/27937>

## ANEXOS

### PARALLAX MEMORY STICK DATALOGGER – HOJA DE DATOS

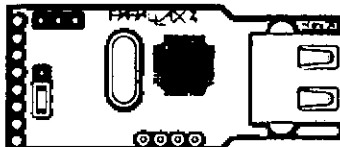
#### Specifications

Symbol	Quantity	Minimum	Typical	Maximum	Units
Vdd	Supply Voltage	4.75	5.0	5.25	V
-	Storage Temperature	-65°	-	150°	C
-	Operating Temperature	0°	-	70°	C
Ivdd	Supply Current (Running)	-	25	-	mA
Ivdd	Supply Current (Standby)	1	2	2	mA

#### Pin Definitions (UART Mode)

Pin	Name	Description
1	Vss	Connects to System Ground
2	RTS#	Request To Send (Connects to MCU CTS)
3	Vdd	Connects to +5V (Regulated)
4	RXD	Receive Data (Connects to MCU TXD)
5	TXD	Transmit Data (Connects to MCU RXD)
6	CTS#	Clear To Send (Connects to MCU RTS)
7	NC	No Connection
8	RI#	Ring Indicator (Making this input low resumes from Suspend)

**Jumper in UART Mode**



#### LED Definitions

Operation	LED Behavior
Power On	Green LED and Red LED flash alternately for 2 seconds Repeated until monitor connects
USB Disk Initialization	Green LED on, Red LED off
USB Disk Ready	Green LED off, Red LED on
USB Disk Removed	Green LED off, Red LED off
Commands From Monitor Port to USB Disk	Green LED off, Red LED flashes
Commands From Monitor Port with USB Disk Removed	Green LED off, Red LED off

## PARALLAX MEMORY STICK DATALOGGER – COMANDOS

Extended ASCII Command for Terminal mode	Shortened Hexadecimal Command for microprocessor mode	Command function	Response
<b>Switching between Shortened and Extended Command sets</b>			
'SCS'<cr>	\$10,\$0D	Switches to the shortened command set	This will return the prompt '>',\$0D to indicate that the device is in shortened command set mode.
'ECS'<cr>	\$11,\$0D	Switches to the extended command set	This will return the prompt 'D:\>',\$0D to indicate that the device is in extended command set mode.
'E'<cr>	'E'<cr>	Echo	This will return 'E',\$0D for synchronisation purposes
'e'<cr>	'e'<cr>	Echo	This will return 'e',\$0D for synchronisation purposes
<b>Responses to indicate if disk is online</b>			
<cr>	\$0D	Check if online	This will return the appropriate prompt or 'no disk' message for the current command set.
Response to Check if online for Extended Command Mode		If no valid disk is found	'No Disk',\$0D
		If a valid disk is found	'D:\>',\$0D
Response to Check if online for Short Command Mode		If no valid disk is found	'ND',\$0D
		If a valid disk is found	'>',\$0D
<b>Directory operations</b>			
'DIR'<cr>	\$01,\$0D	Lists the current directory	A list of file names and directory names are returned. Each entry is terminated by \$0D. A directory entry has <sp>'DIR' after the name and before the \$0D.
'DIR'<sp> <name><cr>	\$01,\$20,<name>,\$0D	Lists the file name followed by the size. Use this before doing a file read to know how many bytes to expect.	\$0D,<name><sp><size in hex(4 bytes) LSB first> \$0D
'DLD'<sp> <name><cr>	\$05,\$20,<name>,\$0D	Delete directory	Deletes the directory <name> from the current directory. <prompt>\$0D
'MKD'<sp> <name><cr>	\$06,\$20,<name>,\$0D	Make directory	Creates a new directory <name> in the current directory. <prompt>\$0D
'CD'<sp> <name><cr>	\$02,\$20,<name>,\$0D	The current directory is changed to the new directory <name>	<prompt>\$0D
'CD'<sp>'.','<cr>	\$02,\$20,\$2E,\$2E,\$0D	Move up one directory level.	<prompt>\$0D
<b>File operations</b>			
'RD'<sp> <name><cr>	\$04,\$20,<name>,\$0D	Read file <name>	This will send back the entire file in binary to the monitor. The size should first be found by using the 'DIR' <sp> <name> <cr> command so that the expected number of bytes is known. <prompt>\$0D
'RDF'<sp> <size in hex(4 bytes MSB first) ><cr>	\$0B,\$20,<size in hex(4 bytes) >,\$0D	Reads the data of <size in hex(4 bytes) > from the current open file.	This will send back the requested amount of data to the monitor. <prompt>\$0D
'DLF'<sp> <name><cr>	\$07,\$20,<name>,\$0D	Delete file <name>	This will delete the file from the current directory and free up the FAT sectors. <prompt>\$0D
'WRF'<sp> <size in hex(4 bytes MSB first) ><cr> <data bytes of size><cr>	\$08,\$20,<size in hex(4 bytes) >,\$0D \$data,\$0D	Writes the data of <size in hex(4 bytes) > to the end of the current open file.	<prompt>\$0D
'OPW'<sp> <name><cr>	\$09,\$20,<name>,\$0D	Opens a file for writing to with 'WRF'	<prompt>\$0D
'OPR'<sp> <name><cr>	\$0E,\$20,<name>,\$0D	Opens a file for reading to with 'RDF'	<prompt>\$0D
'CLF'<sp> <name><cr>	\$0A,\$20,<name>,\$0D	Closes a file for writing.	<prompt>\$0D

## BLUETOOTH HC-06 – CONFIGURACIÓN

### Default:

Slave, 9600 baud rate, N, 8, 1. Pincode 1234

### AT command:

#### 1. Communications Test :

Sent : AT

receive : OK

#### 2. Change baud rate :

Sent : AT+BAUD1

receive : OK1200

Sent : AT+BAUD2

receive : OK2400

1-----1200  
2-----2400  
3-----4800  
4-----9600  
5-----19200  
6-----38400  
7-----57600  
8-----115200

Baud rate setting can be save even power down.

#### 3. Change Bluetooth device name:

Sent : AT+NAMEdevicename

receive : OKname

(devicename is the name you want the device to be , and it will be searched with this name)

Name setting can be save even power down.

#### 4. Change Pincode:

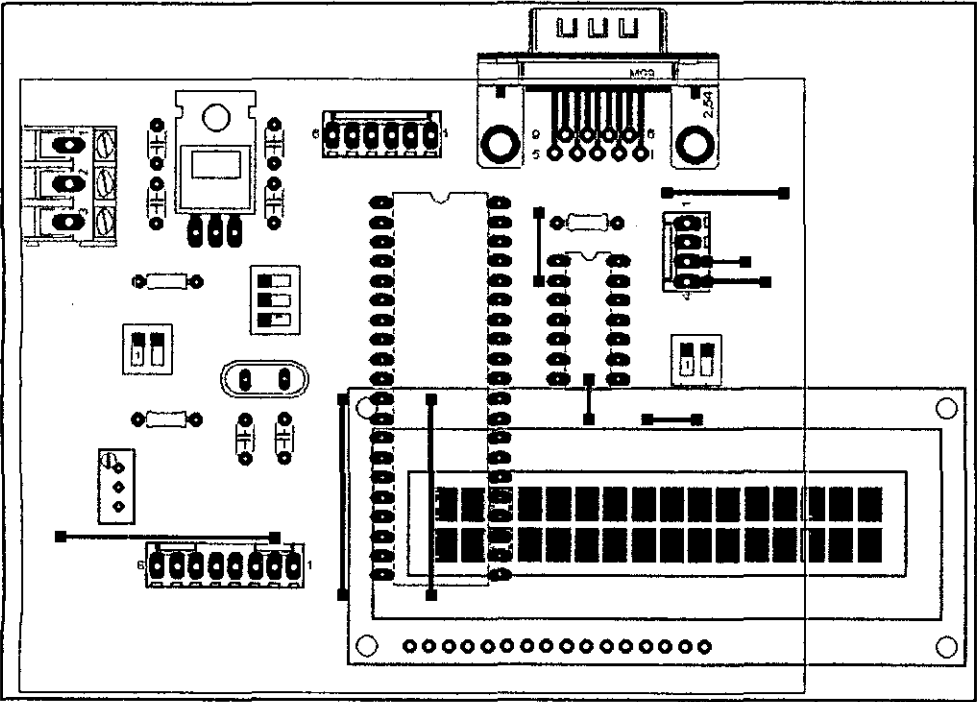
Sent : AT+PINxxxx

receive : OKsetpin

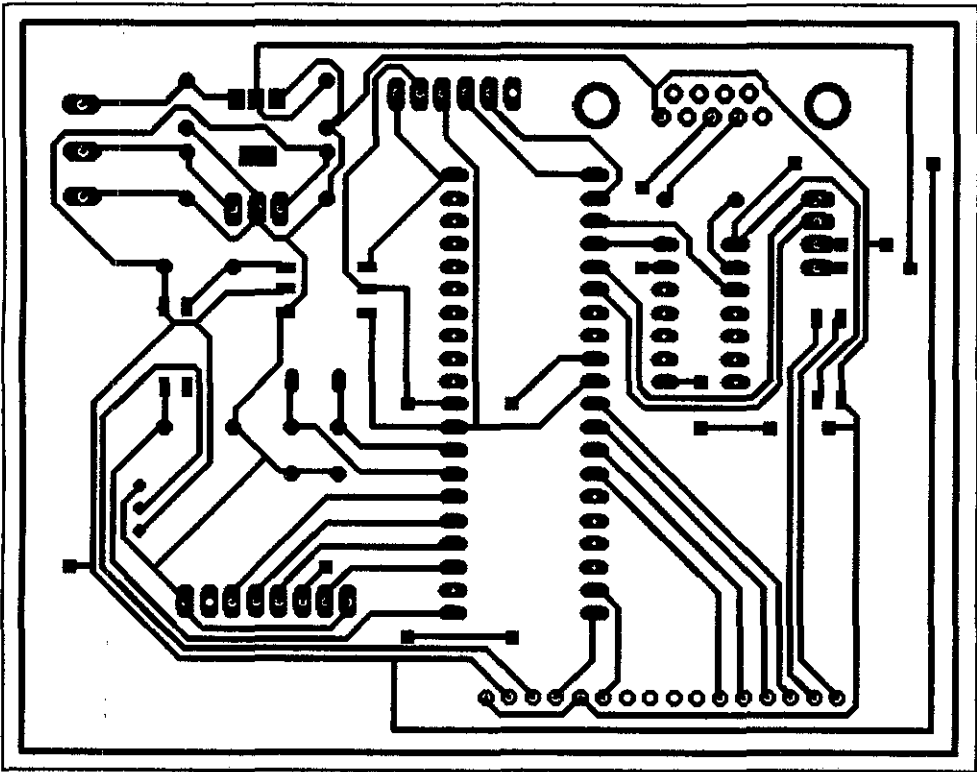
(xxxx is the pin code you set)

Pin code can be save even power down.

**PCB DEL PROTOTIPO – EAGLE**



**Ubicación de los componentes**



**Pistas, vías y agujeros**